



Universidad  
Carlos III de Madrid  
[www.uc3m.es](http://www.uc3m.es)

# TRABAJO FIN DE GRADO

## Automatización de la transformación del modelo de datos de la especificación XBRL en Base de datos

Autor: Yolanda León Román

Titulación: Grado en Ingeniería Informática

Tutores: Elena Castro Galán e Ignacio J. Santos Forner

Leganés, 4 de septiembre 2012



# Agradecimientos

A mi madre por su apoyo continuo



## Contenido

1. MOTIVACIÓN Y OBJETIVOS.....	16
1.1 Misión.....	16
1.2 Motivación .....	16
1.3 Objetivos .....	16
2. PLANTEAMIENTO DEL PROBLEMA.....	17
2.1 Situación histórica .....	17
2.2 ARELLE .....	17
2.3 XML .....	17
2.5 XBRL .....	18
2.5.1 Creación de XBRL.....	18
2.5.2 Qué es XBRL.....	18
2.5.3 Estructura XBRL .....	18
2.6 Estado del arte .....	20
2.7 Situación actual .....	20
2.7.1 Base de datos inicial .....	21
2.8 Solución propuesta.....	22
3. ESTUDIO Y ALCANCE DEL SISTEMA.....	24
3.1 Hito 1 .....	24
3.2 Hito 2 .....	24
3.3 Hito 3 .....	24
4. ANALISIS DE LA SOLUCIÓN TÉCNICA.....	26
4.1 REQUISITOS DEL SISTEMA .....	26
4.1.1 Requisitos Funcionales .....	26
4.1.1.1 Hito 1:.....	27
4.1.1.2 Hito 2:.....	30
4.1.1.3 Hito 3:.....	34
4.1.2 Requisitos no funcionales.....	38
4.2 CASOS DE USO .....	39
4.2.1 Especificación de los casos de uso.....	39
4.2.2 Diagrama de casos de uso .....	41
4.3 IDENTIFICACIÓN DE LOS SUBSISTEMAS .....	42
5. DISEÑO TÉCNICO DEL SISTEMA .....	44

5.1 SELECCIÓN DE LA ARQUITECTURA.....	44
5.2 ARQUITECTURA DE SOPORTE .....	44
5.2.1 Identificación de Subsistemas de Diseño.....	44
5.2.2 Especificación del Entorno Tecnológico.....	44
5.2.2.1 Hardware .....	44
5.2.2.2 Software .....	45
5.2.3 Diseño de Subsistemas de Soporte.....	45
5.2.3.1 Subsistema de Hito 1 .....	46
5.2.3.2 Subsistema de Hito 2 .....	48
5.2.3.3 Subsistema de Hito 3 .....	49
5.2.4 Diagrama de flujo .....	60
5.2.4.1 Subsistema de Hito .....	61
5.2.4.2 Subsistema de Hito 2 .....	62
5.2.4.3 Subsistema de Hito 3 .....	63
5.2.5 Diseño de casos de uso reales .....	67
5.3 DISEÑO FÍSICO DE DATOS .....	67
5.3.1 Diseño del Modelo Físico de Datos .....	67
5.3.2 Elaboración de Especificaciones del Modelo Físico de Datos .....	68
5.4 IMPLEMENTACIÓN DE LA SOLUCIÓN .....	70
5.4.1 Hito 1 .....	70
5.4.1.1 Parte 1: FINREP 2012 .....	70
5.4.1.1.1 Desarrollo del programa xbrl001.vbs.....	71
5.4.1.2 Parte 2: Estudio de otras Taxonomías .....	72
5.4.1.2.1 FINREP 2008 .....	72
5.4.1.2.2 Solvency II.....	72
5.4.1.3 Diagrama de flujo .....	74
5.4.1.4 Manual para el usuario .....	74
5.4.2 Hito 2 .....	74
5.4.2.1 FINREP 2012 .....	74
5.4.2.1 .1 Desarrollando el programa .....	75
5.4.2.2 Solvency II.....	76
5.4.2.2.1 Modificando el programa xbrl002.vbs .....	77
5.4.2.4 Manual para el usuario .....	78
5.4.3 Hito 3 .....	78

5.4.3.1 Solvency II.....	79
5.4.3.1.1 Assertion AS17A .....	79
5.4.3.2 FINREP 2012 .....	81
5.4.3.2.1 Assertion t03_hr01 .....	81
5.4.3.3 Manual de usuario.....	87
6. PRUEBAS DE LA SOLUCIÓN TÉCNICA .....	88
6.1 Pruebas de la solución técnica .....	88
6.1.1 Hito 1 .....	88
6.1.2 Hito 2 .....	92
6.1.3 Hito 3 .....	95
6.2 Pruebas de la solución obtenida.....	99
6.2.1 Hito 1 .....	100
6.2.2 Hito 2 .....	102
6.2.3 Hito 3 .....	103
6.2.3.1 Pruebas generales Solución .....	104
6.2.3.2 Aserción AS17A de Solvency II .....	105
6.2.3.3 Aserción t03-hr01 de FINREP 2012 .....	108
7. EVOLUCIÓN Y RESULTADOS .....	110
7.1 Hito 1 .....	110
7.1.1 FINREP 2012 .....	110
7.1.2 Solvency II.....	112
7.2 Hito 2 .....	113
7.2.1 FINREP 2012 .....	114
7.2.2 Solvency II.....	116
7.3 Hito 3 .....	117
7.3.1 Solvency II.....	117
7.3.2 FINREP 2012 .....	118
8. PLANIFICACIÓN .....	120
9. PRESUPUESTO .....	124
10. CONCLUSIONES Y TRABAJO FUTURO.....	125
11. RESUMEN .....	126
12. ANEXOS .....	129
12.1 GLOSARIO DE TÉRMINOS.....	129
12.2 BIBLIOGRAFIA .....	130

12.3 TAXONOMIAS .....131

12.3.1 FINREP 2008 .....131

12.3.2 FINREP 2012 .....131

12.3.3 Solvency II.....131



## Ilustraciones

Ilustración 1: Ejemplo XML.....	18
Ilustración 2: XBRL Estructurado .....	19
Ilustración 3: Ejemplo de Informe XBRL v2.1.....	19
Ilustración 4: Modelo base de datos .....	22
Ilustración 5: Diagrama de casos de uso .....	42
Ilustración 6: Subsistemas de Análisis .....	43
Ilustración 7: Diagrama de Flujo Hito 2 .....	61
Ilustración 8: Diagrama de Flujo Hito 2 .....	62
Ilustración 9: Diagrama de Flujo H3 general .....	64
Ilustración 10: Diagrama de Flujo H3 tipo 6 .....	65
Ilustración 11: Diagrama de Flujo H3 tipo 7 .....	66
Ilustración 12: FINREP 2012 Conceptos básicos Arelle .....	71
Ilustración 13: FINREP 2012 Dimensiones Arelle .....	71
Ilustración 14: FINREP 2008 Fichero Arelle .....	72
Ilustración 15: Conceptos Básicos en Solvency2 .....	73
Ilustración 16: Dimensiones en Solvency2 .....	73
Ilustración 17: Cabecera de Instancia de FINREP 2012.....	74
Ilustración 18: Contexto de instancia de FINREP 2012 .....	75
Ilustración 19: Hecho de instancia de FINREP 2012 .....	75
Ilustración 20: Cabecera de instancia Solvency II .....	77
Ilustración 21: Contexto de instancia de Solvency II .....	77
Ilustración 22: Hechos de Instancia de Solvency II .....	77
Ilustración 23: Solvency II Aserción AS17A.....	79
Ilustración 24: FINREP 2012 aserción t03_hr01 fórmula y filtros .....	82
Ilustración 25: FINREP 2012 aserción t03_hr01 variables .....	83
Ilustración 26: FINREP 2012 aserción t03_hr01 xml .....	83
Ilustración 27: Aserción FINREP 2006 informe 6610 .....	84
Ilustración 28: FINREP 2012 aserción t03_hr01 modificado.....	85
Ilustración 29: FINREP2012 atributos de dimensión .....	110
Ilustración 30: FINREP2012 dimensiones .....	111
Ilustración 31: FINREP2012 conceptos básicos .....	111
Ilustración 32: FINREP 2012 Relación Dim_Atributo .....	112
Ilustración 33: Solvency II Atributos de dimensión .....	112
Ilustración 34: Solvency II Dimensiones .....	112
Ilustración 35: Solvency II Conceptos básicos .....	113
Ilustración 36: Solvency II Relación Dim_Atributo.....	113
Ilustración 37: FINREP Cabecera .....	114
Ilustración 38: FINREP contexto .....	114
Ilustración 39: FINREP Hechos.....	115
Ilustración 40: FINREP unidad .....	115
Ilustración 41: FINREP Relación contexto atributo de dimensión .....	116
Ilustración 42: Solvency II cabecera .....	116
Ilustración 43: Solvency II contextos .....	116

Ilustración 44: Solvency II hechos .....	117
Ilustración 45: Solvency II unidad .....	117
Ilustración 46: Solvency II Relación del contexto con los atributos de dimensión .....	117
Ilustración 47: Validación Solvency II .....	117
Ilustración 48: Validación FINREP 2012 Filtro 1 .....	118
Ilustración 49: Validación FINREP 2012 Filtro 2 .....	119
Ilustración 50: Validación FINREP 2012 Filtro 3 .....	119
Ilustración 51: Tareas Gantt General .....	120
Ilustración 52: Tareas Gantt Parte 1 .....	121
Ilustración 53: Tareas Gantt Parte 2 .....	121
Ilustración 54: Diagrama de Gantt Parte 1 .....	122
Ilustración 55: Diagrama de Gantt Parte 2 .....	123
Ilustración 56: Diagrama de Gantt Leyenda .....	123

## Tablas

Tabla 1: Plantilla Requisito .....	26
Tabla 2:Requisito RF-01.....	27
Tabla 3:Requisito RF-02.....	27
Tabla 4: Requisito RF-03.....	27
Tabla 5: Requisito RF-04.....	27
Tabla 6: Requisito RF-05.....	28
Tabla 7: Requisito RF-06.....	28
Tabla 8: Requisito RF-07.....	28
Tabla 9: Requisito RF-08.....	29
Tabla 10: Requisito RF-09.....	29
Tabla 11: Requisito RF-1.....	29
Tabla 12: Requisito RF-11.....	29
Tabla 13: Requisito RF-12.....	30
Tabla 14: Requisito RF-13.....	30
Tabla 15: Requisito RF-14.....	30
Tabla 16: Requisito RF-15.....	30
Tabla 17: Requisito RF-16.....	31
Tabla 18: Requisito RF-17.....	31
Tabla 19: Requisito RF-18.....	31
Tabla 20: Requisito RF-19.....	32
Tabla 21: Requisito RF-20.....	32
Tabla 22: Requisito RF-21.....	32
Tabla 23: Requisito RF-22.....	32
Tabla 24: Requisito RF-23.....	33
Tabla 25: Requisito RF-24.....	33
Tabla 26: Requisito RF-25.....	33
Tabla 27: Requisito RF-26.....	33
Tabla 28: Requisito RF-27.....	34
Tabla 29 : Requisito RF-28.....	34
Tabla 30: Requisito RF-29.....	34
Tabla 31: Requisito RF-30.....	34
Tabla 32: Requisito RF-31.....	35
Tabla 33: Requisito RF-32.....	35
Tabla 34: Requisito RF-33.....	35
Tabla 35: Requisito RF-34.....	36
Tabla 36: Requisito RF-35.....	36
Tabla 37: Requisito RF-36.....	36
Tabla 38: Requisito RF-37.....	36
Tabla 39: Requisito RF-38.....	37
Tabla 40: Requisito RF-39.....	37
Tabla 41: Requisito RF-40.....	37
Tabla 42: Requisito RF-41.....	37
Tabla 43: Requisito RF-42.....	38

Tabla 44: Requisito RF-43 .....	38
Tabla 45: Requisito RF-44 .....	38
Tabla 46: Requisito RNF-01 .....	38
Tabla 47 Requisito RNF-02 .....	39
Tabla 48: Plantilla Casos de Uso .....	39
Tabla 49: Caso de uso CU-01 .....	40
Tabla 50: Caso de uso CU-02 .....	40
Tabla 51: Caso de uso CU-03 .....	41
Tabla 52: Plantilla Programa .....	45
Tabla 53: Plantilla Subrutinas .....	45
Tabla 54: Programa xbrl001 .....	46
Tabla 55: Subrutina H1: Rutina.....	47
Tabla 56: Subrutina H1: read_dimensions .....	47
Tabla 57: Programa xbrl002 .....	48
Tabla 58:Subrutina H2: Rutina .....	49
Tabla 59: Subrutina H2: read_documentInstance .....	49
Tabla 60: Programa xbrl003 .....	52
Tabla 61: Subrutina H3: Rutina.....	52
Tabla 62: Subrutina H3: spExFiles.....	52
Tabla 63: Subrutina H3: read_formulas.....	53
Tabla 64: Subrutina H3: initializeVariables .....	53
Tabla 65: Subrutina H3: assertionNameGet .....	53
Tabla 66: Subrutina H3: assertionAnalyze .....	54
Tabla 67: Subrutina H3: replaceString .....	54
Tabla 68: Subrutina H3: readParameter.....	54
Tabla 69: Subrutina H3: readExplicitFilters.....	55
Tabla 70: Subrutina H3: variableHeader.....	55
Tabla 71: Subrutina H3: dealPreconditionVariable .....	55
Tabla 72: Subrutina H3: errorMessage .....	56
Tabla 73: Subrutina H3: replacePrecondition.....	56
Tabla 74: Subrutina H3: evaluatePrecondition .....	56
Tabla 75: Subrutina H3: dealVariable .....	57
Tabla 76: Subrutina H3: dimVariable.....	57
Tabla 77: Subrutina H3: mapping .....	58
Tabla 78: Subrutina H3: isTypeI.....	58
Tabla 79: Subrutina H3: sptypeI .....	58
Tabla 80: Subrutina H3: sptypeII .....	58
Tabla 81: Subrutina H3: sptypeIII .....	58
Tabla 82: Subrutina H3: sptypeIV .....	59
Tabla 83: Subrutina H3: sptypeV .....	59
Tabla 84: Subrutina H3: sptypeVI .....	59
Tabla 85: Subrutina H3: sptypeVII .....	60
Tabla 86: Subrutina H3: replaceFunctMath.....	60
Tabla 87: Relación subsistema H1 .....	67
Tabla 88: Relación subsistema H2 .....	67

Tabla 89: Relación subsistema H3 .....	67
Tabla 90: Modelo base de datos 2 .....	68
Tabla 91: Script Generación de Tablas .....	70
Tabla 92: Plantilla pruebas diseño técnico .....	88
Tabla 93: Prueba PR-TEC-01 .....	88
Tabla 94: Prueba PR-TEC-02 .....	89
Tabla 95: Prueba PR-TEC-03 .....	89
Tabla 96: Prueba PR-TEC-04 .....	89
Tabla 97: Prueba PR-TEC-05 .....	90
Tabla 98: Prueba PR-TEC-06 .....	90
Tabla 99: Prueba PR-TEC-07 .....	91
Tabla 100: Prueba PR-TEC-08 .....	91
Tabla 101: Prueba PR-TEC-09 .....	91
Tabla 102: Prueba PR-TEC-10 .....	92
Tabla 103: Prueba PR-TEC-12 .....	92
Tabla 104: Prueba PR-TEC-12 .....	92
Tabla 105: Prueba PR-TEC-13 .....	93
Tabla 106: Prueba PR-TEC-14 .....	93
Tabla 107: Prueba PR-TEC-15 .....	93
Tabla 108: Prueba PR-TEC-16 .....	94
Tabla 109: Prueba PR-TEC-17 .....	94
Tabla 110: Prueba PR-TEC-18 .....	95
Tabla 111: Prueba PR-TEC-19 .....	95
Tabla 112: Prueba PR-TEC-20 .....	95
Tabla 113: Prueba PR-TEC-21 .....	96
Tabla 114: Prueba PR-TEC-22 .....	96
Tabla 115: Prueba PR-TEC-23 .....	96
Tabla 116: Prueba PR-TEC-24 .....	96
Tabla 117: Prueba PR-TEC-25 .....	97
Tabla 118: Prueba PR-TEC-26 .....	97
Tabla 119: Prueba PR-TEC-27 .....	98
Tabla 120: Prueba PR-TEC-28 .....	98
Tabla 121: Prueba PR-TEC-29 .....	98
Tabla 122: Prueba PR-TEC-30 .....	99
Tabla 123: Plantilla Prueba solución .....	99
Tabla 124: Prueba PR-SOL-01 .....	100
Tabla 125: Prueba PR-SOL-2 .....	100
Tabla 126: Prueba PR-SOL-03 .....	101
Tabla 127: Prueba PR-SOL-04 .....	101
Tabla 128: Prueba PR-SOL-05 .....	102
Tabla 129: Prueba PR-SOL-06 .....	102
Tabla 130: Prueba PR-SOL-07 .....	102
Tabla 131: Prueba PR-SOL-08 .....	103
Tabla 132: Prueba PR-SOL-09 .....	103
Tabla 133: Prueba PR-SOL-10 .....	104

Tabla 134: Prueba PR-SOL-11 .....	104
Tabla 135: Prueba PR-SOL-12 .....	105
Tabla 136: Prueba PR-SOL-13 .....	106
Tabla 137: Prueba PR-SOL-14 .....	106
Tabla 138: Prueba PR-SOL-15 .....	106
Tabla 139: Prueba PR-SOL-16 .....	106
Tabla 140: Prueba PR-SOL-17 .....	107
Tabla 141: Prueba PR-SOL-18 .....	107
Tabla 142: Prueba PR-SOL-19 .....	107
Tabla 143: Prueba PR-SOL-20 .....	108
Tabla 144: Prueba PR-SOL-21 .....	108
Tabla 145: Prueba PR-SOL-22 .....	109
Tabla 146: Prueba PR-SOL-23 .....	109
Tabla 147: Presupuesto del Proyecto .....	124
Tabla 148: Glosario de Términos .....	130



## 1. MOTIVACIÓN Y OBJETIVOS

### 1.1 Misión

Este trabajo final de grado consiste en la transformación a un lenguaje gestor de base de datos unos documentos instancia XBRL. Para empezar se analizará y estudiará como trata los datos este lenguaje basado en XML. Este trabajo complementa un proyecto más grande de investigación para una tesis doctoral.

XBRL (*eXtensible Business Reporting Language*), es un lenguaje basado, como se ha comentado anteriormente, en XML que sirve para la automatización en el intercambio de información financiera.

### 1.2 Motivación

La motivación por hacer este trabajo final es grado es que durante toda la carrera la parte que más me ha interesado es la parte de base de datos, tanto diseño, creación como administración. Este proyecto como se ha comentado antes, trata de pasar a un lenguaje de base de datos otro lenguaje llamado XBRL, no muy conocido fuera del entorno financiero o económico. La oportunidad de conocer un lenguaje nuevo y poco conocido fue algo que me resulto atractivo desde un primer momento.

### 1.3 Objetivos

Partimos de unas taxonomías utilizadas en el lenguaje XBRL, como pueden ser FINREP 2012, FINREP 2008 y Solvency II y el objetivo final es la creación automatizada de cualquiera de estas taxonomías a un lenguaje gestor de base de datos.

Entendido de donde partimos y a donde queremos llegar el primer objetivo que se plantea es entender el lenguaje XBRL y tener un pequeño conocimiento del lenguaje de Microsoft Visual Basic Script, que será en el que se desarrollarán los programas necesarios para cumplir el objetivo final.

Después de este objetivo inicial se definen tres fases:

1. Primera Fase: Creación de inserciones de los metadatos en base de datos, a partir de fichero plano obtenido por la herramienta Arelle de código abierto.  
Mediante el lenguaje Visual Basic Script obtener los metadatos para la creación de los conceptos básicos, dimensiones, atributos y relación dimensión-atributo.
2. Segunda Fase: Creación de inserciones de los metadatos en base de datos, a partir del fichero instancia en XBRL de la taxonomía.  
Mediante el lenguaje Visual Basic Script acceder al fichero instancia XBRL, y extraer los contextos y hechos, y las relaciones de estos con dimensiones y conceptos básicos.
3. Tercera Fase: Creación y validación de las aserciones de la base de datos, a partir de un fichero plano obtenido con la herramienta Arelle de código abierto.  
Mediante el lenguaje Visual Basic Script generar un programa de Transact SQL que desarrolle la aserción de la taxonomía para analizar y validar los datos introducidos en la base de datos.



## 2. PLANTEAMIENTO DEL PROBLEMA

### 2.1 Situación histórica

Para entender la creación y el uso que actualmente está teniendo el lenguaje de informes financieros XBRL, nos vamos a remontar a su creación:

A principios de los años 70 la necesidad de enviar informes de cuentas y estados no solo entre bancos si no entre empresas también empezó a crecer. Esto de enviar informes era realmente sencillo, se envía un fichero de texto y no había mayor complicación, eso si la mayoría de las veces estos ficheros llegaban con multitud de errores sintácticos diferentes, por ejemplo con caracteres extraños o errores semánticos, no había forma de verificar semánticamente el informe.

Más tarde empezó a surgir varios estándares. Uno de estos estándares es el llamado estándar SDMX (Statistical Data and Metadata Exchange) patrocinado por el Banco Internacional de Pago<sup>9</sup>, el Banco Central Europeo (BCE<sup>10</sup>) , el Organismo para la Cooperación y el Desarrollo Económico(OCDE<sup>11</sup>) y el EuroStat<sup>12</sup>.

SDMX<sup>13</sup> es un conjunto de estándares para facilitar el intercambio de datos y metadatos estadísticos utilizando tecnologías de información, sus principales objetivos son: establecer un conjunto de estándares reconocidos garantizando un acceso y manejabilidad simple y asegurar que los datos vayan acompañados de sus metadatos.

En el año 2000 surge XBRL, un estándar para el envío de información financiera y esto poco a poco ha ido cambiando la idea de estos informes, con XBRL se ha conseguido validar en origen y en destino los datos financieros para que estos no fueran problema de error. Sin embargo aparece una parte complicada de este lenguaje para el envío y la recepción de informes, que muy poca gente conoce y sabe manejar este tipo de lenguaje y este tipo de informes.

### 2.2 ARELLE

Arelle es un proyecto destinado a proporcionar a la comunidad XBRL una plataforma fácil de usar para tratar XBRL. Es un marco robusto y compacto que se usa como aplicación de escritorio.<sup>1</sup>

Con Arelle puede ver la instancia de la base de datos en un formato, gracias a su aplicación, fácil y sencillo de comprender. Arelle trata los ficheros xbrl y los transforma en ficheros de texto plano, de manera que leer estos documentos XBRL se vuelve más sencillo.

### 2.3 XML

XML o por sus siglas Extensible Markup Language, es un simple formato basado en texto para representar información estructurada. Fue derivado del lenguaje SGML (Standard Generalized Markup Language), para adecuarse más al uso de la web. Hoy en día el lenguaje XML se utiliza para todo tipo de intercambio de información.<sup>2</sup>

Este lenguaje está compuesto por etiquetas como puedes ser:

<id></id> o también <id />

Es en lenguaje muy similar a HTML, pero XML sirve para enviar datos y no puede contener errores. Al igual que en HTML las etiquetas puedes estar anudadas.

```
<book>
  <title> Harry Potter and the Philosopher`s Stone </title>
  <autor> J.K. Rowling</autor>
</book>
```

Ilustración 1: Ejemplo XML

## 2.5 XBRL

### 2.5.1 Creación de XBRL

En 1998 Charles Hoffman empezó a investigar con XML para el desarrollo de informes en formato electrónico de información financiera, consiguió que el comité del AICPA (American Institute of Certified Public Accountants) financiara el proyecto, haciendo que el XML fuera considerado importante para la profesión contable. El código se empezó a llamar XFRML por las siglas eXtensible Financial Reporting Markup Language y en el año 2000 cambio su nombre a XBRL por las siglas eXtensible Business Reporting Language y presentó la primera especificación para empresas estadounidenses de estados financieros.<sup>3</sup>

Este cambio de nombre se debe aunque actualmente el mayor avance de XBRL esta surgiendo en el mundo de la información financiera ya que en año 2000, se comprobó que XBRL no sólo servía para información financiera sino también para cualquier otro tipo de informes económicos.

### 2.5.2 Qué es XBRL

XBRL es un estándar libre de XML, es decir, que está disponible públicamente, desarrollado para cualquier usuario y es libre de pago.

XBRL se utiliza para la comunicación electrónica de datos financieros, se define como un diccionario de etiquetas que se pueden aplicar a cada elemento. XBRL está basado en **Taxonomías** que son las reglas que tienen que cumplir los elementos o etiquetas para procesar los datos como si fuera un diccionario de datos. Una taxonomía debe ser aceptada universalmente, y debe establecer los criterios para asignar los datos del documento de instancia a los datos propios del informe, validándose tanto en origen como en destino.

### 2.5.3 Estructura XBRL

XBRL es una versión potente y flexible de XML, permite aplicar etiquetas identificativas que proporcionan una amplia información sobre el elemento.

XBRL utiliza la semántica del XML para expresar significado lo cual es muy importante para los negocios y los informes financieros.

Las taxonomías XBRL se expresan en meta-datos, que son las normas o los datos que tiene que incluir una determinada información, comunes en los diferentes reportes de negocio. Los usuarios pueden añadir meta-datos según se vayan necesitando.

En XBRL se pueden añadir reglas de negocio o formular que se deben cumplir para validar un informe o un documento instancia XBRL, lo bueno del XBRL es que cada usuario puede escribir sus propias reglas o validaciones.

XBRL es un lenguaje estructurado, de manera que tanto para el usuario como para la máquina va a ser más fácil poder llegar a obtener la información deseada. XBRL se estructura de la misma manera que una página en HTML, de manera que puedas llegar a la información que quieras sin error. En la imagen siguiente se ve como puede ser la estructura de un lenguaje XBRL, de esta manera podemos acceder al valor del año 2006 o del 2005 o de cualquier otro dato sin posibilidad de error y de manera ordenada y detallada.

```
<Inventory>
  <ConsistsOf>produce purchased for resale and supplies</ConsistsOf>
  <StatedAt>lower of cost or market</StatedAt>
  <ValuationMethod>FIFO</ValuationMethod>
  <Value2006>$45,594</Value2006>
  <Value2005>$34,456</Value2005>
</Inventory>
```

Ilustración 2: XBRL Estructurado

Para tratar los informes XBRL primero se mostrará como se ha definido el elemento <xbrl> que se muestra en la siguiente ilustración:<sup>15</sup>

```
<xbrl xmlns="http://www.xbrl.org/2003/instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:link="http://www.xbrl.org/2003/linkbase"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ci="http://www.xbrl.org/sp/general/2005/taxonomia-pgc-2005"
  xsi:schemaLocation="
    http://www.xbrl.org/sp/general/2005/pgc-2005
    http://www.xbrl.org.es/general/2005/pgc-2005.xsd">
  <link:schemaRef xlink:type="simple"
    xlink:href="http://www.xbrl.org.es/general/2005/pgc-2005.xsd"/>
  <ci:activo precision="3" unitRef="u1" contextRef="c1">727</ci:activo>
  <!-- ... otros elementos de la taxonomía con sus valores ... -->
  <ci:pasivo precision="3" unitRef="u1" contextRef="c1">727</ci:pasivo>
  <context id="c1"><!-- ... --></context>
  <unit id="u1"><!-- ... --></unit>
</xbrl>
```

Ilustración 3: Ejemplo de Informe XBRL v2.1

EL informe XBRL contiene las definiciones de los espacios de nombre que se utilizaran para el resto del documento.

El elemento <link:schemaRef> contiene el esquema a utilizar, en este caso es uno pero podrían ser varios y de diferentes taxonomías.

Después tenemos las entradas de en este caso los elementos <ci:activo y <ci:pasivo, estas serán conceptos de la instancia con sus parámetros de precisión, unidad, y contexto, lo que forma lo que más adelante llamaremos **hecho**.

A continuación se encuentra el contexto <context>, este atributo relaciona el elemento en el que nos encontramos con el contexto dimensional, como dimensión siempre suelen existir entidad y periodo.

Y por ultimo nos encontramos el elemento <unit>, este elemento representa la unidad monetaria que trata la instancia.

A llegado el momento de hablar de las taxonomías mencionadas anteriormente y una de las bases de este proyecto. Una de las definiciones que podemos darle a taxonomía es el código o lenguaje de la comunicación, para que ambas partes, emisor y receptor entiendan el mensaje.

Una taxonomía contiene esquemas que es el conjunto de elementos que aparecen en los informes y en su estructura.

Todos los elementos que se definan en una taxonomía deben tener un periodo para identificar la relación que el concepto o elemento tiene con la coordenada tiempo.

## 2.6 Estado del arte

Existen muchas aplicaciones y programas que generan lenguaje XBRL, es decir, que transforman un fichero a XBRL, como puede ser Microsoft Excel.

Mi trabajo como parte de la tesis doctoral, consiste en coger esas taxonomías o instancias en lenguaje XBRL y transformarlo a un lenguaje de base de datos, en mi caso SQL, de manera que cuando se tenga que tratar el lenguaje XBRL podamos usar este de manera más fácil, ya que estará en SQL, con sus tablas, atributos y relaciones.

Viendo esto llegamos a la conclusión de que para pasar “algo” a XBRL existe programas y empresas que se encargan de ello, pero el proceso inverso que sería pasar XBRL a otro formato en que nos sea más cómodo trabajar no está realizado de manera gratuita y fácil, existen entidades XBRL que pueden ayudar a este proceso. El trato del lenguaje XBRL, se dará por las entidades reguladoras y los bancos centrales de cada país. El tratar este lenguaje es un proceso muy lento y se necesita que sea ágil ya que estos documentos en XBRL traerán datos importantes sobre estado de cuentas de bancos y entidades y necesitan ser tratados con rapidez.

En el año 2014 entrará en varias directivas europeas para el envío de informes económicos en XBRL, y actualmente existen muy pocos profesionales que sepan tratar informes XBRL, con lo cual será de mucha utilidad un programa de transformación de XBRL a SQL. Se elige SQL por que la mayoría de los bancos tiene ya profesionales que manejen este gestor de base de datos.

## 2.7 Situación actual

Como ya se ha comentado en momentos anteriores, este proyecto es parte de un proyecto mayor para una tesis doctoral, como consecuencia de esto, al inicial este proyecto ya se tiene

cierta información, ciertos programas y el diseño de la base de datos. Estos programas e información están realizados para la taxonomía de [FINREP 2008](#) y 2012.

La base de datos definida y obtenida del trabajo de la tesis doctoral, será totalmente útil para este proyecto ya que aunque las taxonomías tratadas son diferentes todas estas forman parte de XBRL y XBRL almacena ciertos datos para cualquier instancia o taxonomía.

Las taxonomías son lo que cambian, ese diccionario comentado anteriormente o ese código o lenguaje de comunicación, y por ello se necesita crear unos programas lo más genéricos posible para tratar así cualquier taxonomía de XBRL.

#### ***2.7.1 Base de datos inicial***

El modelo de la estructura XBRL en base de datos es el siguiente, como se muestra en la ilustración 4:

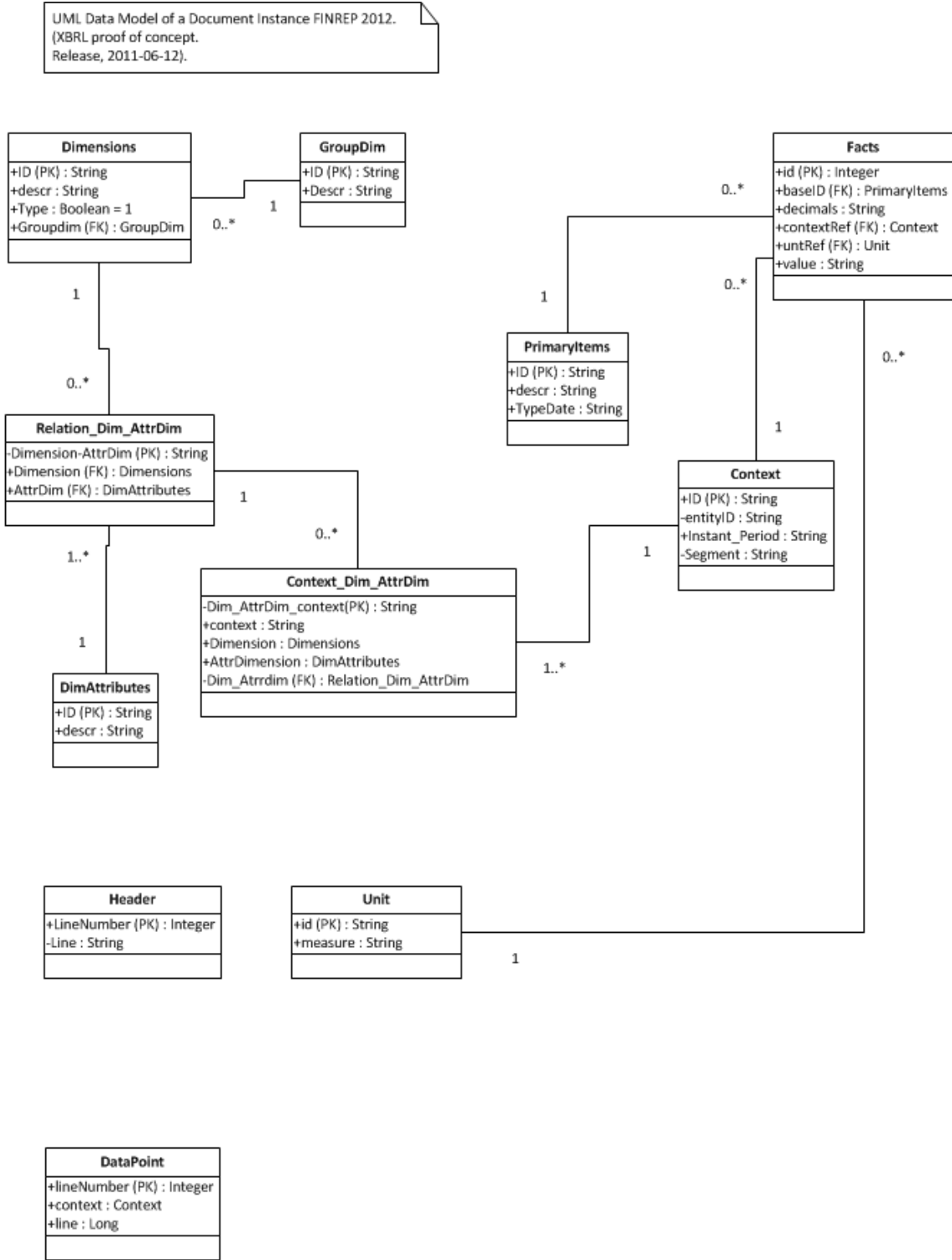


Ilustración 4: Modelo base de datos

### 2.8 Solución propuesta

Visto el problema generado en el apartado de estado del arte, gracias a que no existe una tecnología que convierta los informes en XBRL a un sistema gestor de base de datos, se empezó a desarrollar la tesis doctoral mencionada anteriormente que estudia el lenguaje y los informes XBRL y desarrolla una transformación de estos informes XBRL a un Modelo Multidimensional. Este proyecto se encuentra en el marco de esta tesis que consiste en la

realización de una serie de programas que transformen los informes XBRL a datos, tablas y aserciones en SQL. Estos programas deberán ser lo más genéricos posibles para así tratar diferentes taxonomías.

El alcance de este proyecto será la transformación de las taxonomías de FINREP 2012 y SOLVENCY II y el estudio de cómo actúan los programas para otras taxonomías y el coste de que los programas traten todo tipo de taxonomías XBRL. Este proyecto como se comentó anteriormente consta de tres fases o hito y el alcance de cada hito es el siguiente:

Hito 1: Creación e inserción de todas las dimensiones, atributos de dimensión, conceptos básicos y relaciones de dimensión-atributo (meta datos de las taxonomías).

Hito 2: Creación e inserción de los contextos, hechos, cabecera, y relaciones, cargando el documento instancia XBRL o informes XBRL.

Hito 3: Creación y validación de las aserciones para la validación y el tratamiento de los datos introducidos anteriormente.

### 3. ESTUDIO Y ALCANCE DEL SISTEMA

#### 3.1 Hito 1

En este primer hito del proyecto se va a trabajar con diferentes taxonomías, estudiando cómo podemos tratarlas para obtener los datos necesarios para poder transformarlas a SQL.

Este hito se divide en dos partes, la primera parte estudiará y transformará la taxonomía FINREP 2012, y la segunda parte tratará de un estudio de diferentes taxonomías y ver como estas actúan con el programa generado para la primera parte.

En este hito partimos de un programa desarrollado para FINREP 2008 por la tesis doctoral, el cual tendremos que modificar y mejorar su rendimiento para poder tratar la taxonomía de FINREP 2012.

El primer paso de esta parte para cualquier taxonomía que tratemos será pasar la instancia de la taxonomía por **Arelle** para así poder tratar los datos con más facilidad. De este primer paso obtenemos un fichero en formato plano en el que están los conceptos básicos, dimensiones y atributos de dimensión.

La finalidad de este programa es que tratando el fichero obtenido con Arelle se cree un fichero de inserciones en el que se obtenga los datos contenidos en este fichero Arelle.

#### 3.2 Hito 2

Este segundo hito vamos a trabajar con aquellas taxonomías en las que en el hito 1 se consiguieron los resultados esperados, con lo cual el Hito 2 y como consecuencia el Hito 3, trataran las taxonomías de FINREP 2012 y Solvency II.

Puesto que tratamos dos taxonomías, este hito se dividirá en dos partes, una para la taxonomía de FINREP 2012 y otra para la taxonomía de Solvency II.

Al igual que en la parte anterior, se parte de un programan creado para la taxonomía de FINREP 2008 en su informe 6610, el cual integraremos y mejoraremos para nuestra taxonomías.

En esta parte del proyecto trataremos directamente con la instancia de la taxonomía que está en formado XBRL, en el que se encuentran los contextos y hechos de la taxonomía.

La finalidad y el alcance de este hito será que tratando con el fichero instancia de la taxonomía obtengamos todas las inserciones para la cabecera, el contexto, los hechos y las relaciones entre el contexto y las dimensiones.

#### 3.3 Hito 3

En el tercer hito vamos a trabajar con las taxonomías ya tratadas anteriormente que son FINREP 2012 y Solvency II.

Este hito al igual que los anteriores se dividirá en dos partes, una para cada taxonomía.

En este hito partimos de un programa en VBS, realizado para FINREP 2008 en su informe 6610, en el cual se tratan las formulas y las validaciones diferenciándolas por tipos. Para el desarrollo



de esta parte se tendrá que modificar dicho programa y añadir los tipos que se consideren necesarios.

En esta última parte del proyecto usaremos Arelle para pasar a texto plano las formulas de la taxonomía y así poder usar estas formulas con más facilidad.

Las taxonomías XBRL viene con una serie de aserción de validación que se deben cumplir para asegurarnos de que la taxonomía está correcta. La finalidad y el alcance de este hito es la creación del programa SQL necesario para desarrollar estas aserciones y así validar la taxonomía. Trataremos las aserciones de la manera más general posible para que así se pueda adecuar con otras aserciones del mismo tipo.

## 4. ANALISIS DE LA SOLUCIÓN TÉCNICA

### 4.1 REQUISITOS DEL SISTEMA

En este apartado se definirán las directrices técnicas y de gestión, así como los requisitos de usuario en los que se basará el posterior diseño y desarrollo.

A continuación se mostrará el formato de tabla que van a seguir los requisitos y el significado y valor de cada campo:

- **ID (identificador):** permite identificar unívocamente a un requisito concreto. Sigue el formato RX-YY-HZ, donde X es el tipo de requisito, YY el número de requisito.

Los posibles valores para X son:

- **F:** Funcionales (RF-YY)
- **NF:** No Funcionales (RNF-YY)
- **Nombre:** define al requisito de manera breve.
- **Descripción:** describe de manera clara y concisa pero suficientemente completa el requisito.
- **Necesidad:** determina lo importante que es el requisito para que el sistema funcione adecuadamente, distinguiendo desde funcionalidades imprescindibles hasta superfluas. Sus posibles valores son: **baja, media y alta.**
- **Prioridad:** determina la prioridad que debe tomarse para implementar un requisito. Sus posibles valores son: **baja, media y alta.**
- **Riesgo:** estimación del riesgo que supone implementar un requisito. Está relacionado con la dificultad para implementarlo. Sus posibles valores son: **bajo, medio y alto.**
- **Estabilidad:** estimación de la probabilidad de que un requisito cambie a lo largo del proyecto. Sus posibles valores son: **baja, media y alta.**

<b>ID</b>	RX-YY		
<b>Nombre</b>	Nombre		
<b>Descripción</b>	Descripción del requisito.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 1: Plantilla Requisito

#### 4.1.1 Requisitos Funcionales

En este apartado se muestran los requisitos funcionales, estos requisitos han sido divididos dependiendo de los hitos, para mejorar la búsqueda a la hora de consultarlos. Algunos

requisitos son muy parecidos pero se ha decidido ponerlos en cada hito, ya que cada hito generará un programa VBS diferente que generarán una solución que dependerá de la anterior y debe quedar claro que puede hacer cada uno de estos programas:

#### 4.1.1.1 Hito 1:

<b>ID</b>	RF-01		
<b>Nombre</b>	Introducir <b>Fichero Dimensión</b>		
<b>Descripción</b>	Se podrá introducir <b>el fichero de dimensiones</b> de la taxonomía que quiere estudiar.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 2:Requisito RF-01

<b>ID</b>	RF-02		
<b>Nombre</b>	Introducir <b>Fichero Log</b> en el hito 1.		
<b>Descripción</b>	Se podrá introducir el nombre <b>del fichero log</b> .		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 3:Requisito RF-02

<b>ID</b>	RF-03		
<b>Nombre</b>	Introducir <b>Fichero Inserciones</b> en el hito 1.		
<b>Descripción</b>	Se podrá introducir el nombre <b>del fichero de inserciones</b> .		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 4: Requisito RF-03

<b>ID</b>	RF-04		
<b>Nombre</b>	Introducir <b>Base de datos</b> en el hito 1.		
<b>Descripción</b>	Se podrá introducir el nombre <b>de la base de datos</b>		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 5: Requisito RF-04

<b>ID</b>	RF-05		
<b>Nombre</b>	<b>Fi chero Log</b> en el hito 1.		
<b>Descripción</b>	<p>Contendrá la información relacionado con la ejecución del programa:</p> <ul style="list-style-type: none"> <li>• Nombre del programa que ejecuta</li> <li>• Fecha de inicio de la ejecución</li> <li>• Versión de <b>VBS</b></li> <li>• <b>Fichero Dimensión</b></li> <li>• <b>Fichero log</b></li> <li>• <b>Fichero Inserciones</b></li> <li>• Nombre de la base de datos</li> <li>• Fecha de fin de ejecución</li> </ul>		
<b>Necesidad</b>	Alta/ <b>Media</b> /Baja	<b>Riesgo</b>	Alto/Medio/ <b>Bajo</b>
<b>Prioridad</b>	Alta/ <b>Media</b> /Baja	<b>Estabilidad</b>	Alta/ <b>Media</b> /Baja

Tabla 6: Requisito RF-05

<b>ID</b>	RF-06		
<b>Nombre</b>	Inserción de <b>Conceptos Básicos</b>		
<b>Descripción</b>	Se deberán introducir al fichero de inserciones la inserción de los todos los conceptos básicos del <b>fichero de dimensión</b> .		
<b>Necesidad</b>	<b>Alta</b> /Media/Baja	<b>Riesgo</b>	<b>Alto</b> /Medio/Bajo
<b>Prioridad</b>	<b>Alta</b> /Media/Baja	<b>Estabilidad</b>	Alta/Media/ <b>Baja</b>

Tabla 7: Requisito RF-06

<b>ID</b>	RF-07		
<b>Nombre</b>	Inserción de <b>Dimensiones</b>		
<b>Descripción</b>	Se deberán introducir al fichero de inserciones la inserción de los todas las dimensiones del <b>fichero de dimensión</b> .		
<b>Necesidad</b>	<b>Alta</b> /Media/Baja	<b>Riesgo</b>	<b>Alto</b> /Medio/Bajo
<b>Prioridad</b>	<b>Alta</b> /Media/Baja	<b>Estabilidad</b>	Alta/Media/ <b>Baja</b>

Tabla 8: Requisito RF-07

<b>ID</b>	RF-08		
<b>Nombre</b>	Inserción de <b>Atributos de dimensión</b>		
<b>Descripción</b>	Se deberán introducir al fichero de inserciones la inserción de los todos los atributos de dimensión del <b>fichero de dimensión</b> .		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 9: Requisito RF-08

<b>ID</b>	RF-09		
<b>Nombre</b>	Inserción de Relaciones de dimensión		
<b>Descripción</b>	Se deberán introducir al fichero de inserciones la relación de las <b>dimensiones</b> con sus <b>atributos de dimensión</b> del fichero de dimensión.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 10: Requisito RF-09

<b>ID</b>	RF-10		
<b>Nombre</b>	Inserción de base de datos en el hito 1.		
<b>Descripción</b>	Se deberán introducir al fichero de inserciones el nombre de la <b>base de datos</b> a utilizar.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 11: Requisito RF-1

<b>ID</b>	RF-11		
<b>Nombre</b>	Leer Fichero de Dimensión		
<b>Descripción</b>	Leerá el fichero de dimensión en <b>texto plano</b> .		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 12: Requisito RF-11

<b>ID</b>	RF-12		
<b>Nombre</b>	Crear Fichero de Inserción en el hito 1.		
<b>Descripción</b>	Crearé el fichero de inserción en formato <b>SQL</b> .		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 13: Requisito RF-12

<b>ID</b>	RF-13		
<b>Nombre</b>	Crear Fichero Log en el hito 1.		
<b>Descripción</b>	Crearé el fichero log.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 14: Requisito RF-13

#### 4.1.1.2 Hito 2:

<b>ID</b>	RF-14		
<b>Nombre</b>	Introducir Fichero Instancia		
<b>Descripción</b>	Se podrá introducir el fichero instancia en formato XBRL de la taxonomía que se quiere estudiar.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 15: Requisito RF-14

<b>ID</b>	RF-15		
<b>Nombre</b>	Introducir <b>Fichero Log</b> en el hito 2		
<b>Descripción</b>	Se podrá introducir el nombre <b>del fichero log</b> .		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 16: Requisito RF-15

<b>ID</b>	RF-16		
<b>Nombre</b>	Introducir <b>Fichero Inserciones</b> en el hito 2.		
<b>Descripción</b>	Se podrá introducir el nombre <b>del fichero de inserciones</b> .		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 17: Requisito RF-16

<b>ID</b>	RF-17		
<b>Nombre</b>	Introducir <b>Base de datos</b> en el hito 2.		
<b>Descripción</b>	Se podrá introducir el nombre <b>de la base de datos</b>		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 18: Requisito RF-17

<b>ID</b>	RF-18		
<b>Nombre</b>	<b>Fi chero Log</b> en el hito 2.		
<b>Descripción</b>	<p>Contendrá la información relacionado con la ejecución del programa:</p> <ul style="list-style-type: none"> <li>• Nombre del programa que ejecuta</li> <li>• Fecha de inicio de la ejecución</li> <li>• Versión de VBS</li> <li>• Fichero Instancia</li> <li>• Fichero log</li> <li>• Fichero Inserciones</li> <li>• Nombre de la base de datos</li> <li>• Fecha de fin de ejecución</li> </ul>		
<b>Necesidad</b>	Alta/ <b>Media</b> /Baja	<b>Riesgo</b>	Alto/Medio/ <b>Bajo</b>
<b>Prioridad</b>	Alta/ <b>Media</b> /Baja	<b>Estabilidad</b>	Alta/ <b>Media</b> /Baja

Tabla 19: Requisito RF-18

<b>ID</b>	RF-19		
<b>Nombre</b>	Inserción de base de datos en el hito 2.		
<b>Descripción</b>	Se deberán introducir al fichero de inserciones el nombre de la base de datos a utilizar.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 20: Requisito RF-19

<b>ID</b>	RF-20		
<b>Nombre</b>	Leer Fichero Instancia		
<b>Descripción</b>	Leerá el fichero instancia en formato XBRL.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 21: Requisito RF-20

<b>ID</b>	RF-21		
<b>Nombre</b>	Crear Fichero de Inserción en el hito 2.		
<b>Descripción</b>	Crearé el fichero de inserción en formato <b>SQL</b> .		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 22: Requisito RF-21

<b>ID</b>	RF-22		
<b>Nombre</b>	Crear Fichero Log en el hito 2.		
<b>Descripción</b>	Crearé el fichero log.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 23: Requisito RF-22



<b>ID</b>	RF-23		
<b>Nombre</b>	Inserción de la <b>cabecera</b> .		
<b>Descripción</b>	Se deberán introducir al fichero de inserciones SQL las inserciones de la información relacionada con la cabecera del fichero instancia.		
<b>Necesidad</b>	Alta/ <b>Media</b> /Baja	<b>Riesgo</b>	Alto/ <b>Medio</b> /Bajo
<b>Prioridad</b>	Alta/ <b>Media</b> /Baja	<b>Estabilidad</b>	Alta/Media/ <b>Baja</b>

Tabla 24: Requisito RF-23

<b>ID</b>	RF-24		
<b>Nombre</b>	Inserción del <b>contexto</b> .		
<b>Descripción</b>	Se deberán introducir al fichero de inserciones SQL las inserciones de todos los contextos del fichero instancia.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 25: Requisito RF-24

<b>ID</b>	RF-25		
<b>Nombre</b>	Inserción de la relación contexto-dimensión-atributo		
<b>Descripción</b>	Se deberán introducir al fichero de inserciones SQL las inserciones la relación de cada atributo de dimensión con su dimensión a su contexto según el fichero instancia.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 26: Requisito RF-25

<b>ID</b>	RF-26		
<b>Nombre</b>	Inserción de la <b>unidad</b>		
<b>Descripción</b>	Se deberán introducir al fichero de inserciones SQL la unidad con la que se trabaja en el fichero instancia		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 27: Requisito RF-26

<b>ID</b>	RF-27		
<b>Nombre</b>	Inserción del <b>Hecho</b>		
<b>Descripción</b>	Se deberán introducir al fichero de inserciones SQL las inserciones de todos los hechos contenidos en el fichero instancia.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 28: Requisito RF-27

#### 4.1.1.3 Hito 3:

<b>ID</b>	RF-28		
<b>Nombre</b>	Introducir Fichero aserción		
<b>Descripción</b>	Se podrá introducir el <b>fichero aserción</b> que se quiera estudiar de la taxonomía.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 29 : Requisito RF-28

<b>ID</b>	RF-29		
<b>Nombre</b>	Introducir Fichero log en el hito 3		
<b>Descripción</b>	Se podrá introducir el nombre del fichero log.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 30: Requisito RF-29

<b>ID</b>	RF-30		
<b>Nombre</b>	Introducir nombre de <b>Fichero Procedure</b> .		
<b>Descripción</b>	Se podrá introducir el nombre del fichero procedure de SQL.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 31: Requisito RF-30

<b>ID</b>	RF-31		
<b>Nombre</b>	Introducir nombre de <b>Fichero Validation SQL</b> .		
<b>Descripción</b>	Se podrá introducir el nombre del fichero de validación.		
<b>Necesidad</b>	<b>Alta</b> /Media/Baja	<b>Riesgo</b>	<b>Alto</b> /Medio/Bajo
<b>Prioridad</b>	<b>Alta</b> /Media/Baja	<b>Estabilidad</b>	Alta/Media/ <b>Baja</b>

Tabla 32: Requisito RF-31

<b>ID</b>	RF-32		
<b>Nombre</b>	Introducir nombre base de datos.		
<b>Descripción</b>	Se podrá introducir el nombre de la base de datos.		
<b>Necesidad</b>	<b>Alta</b> /Media/Baja	<b>Riesgo</b>	<b>Alto</b> /Medio/Bajo
<b>Prioridad</b>	<b>Alta</b> /Media/Baja	<b>Estabilidad</b>	Alta/Media/ <b>Baja</b>

Tabla 33: Requisito RF-32

<b>ID</b>	RF-33		
<b>Nombre</b>	<b>Fi chero Log</b> en el hito 3.		
<b>Descripción</b>	<p>Contendrá la información relacionado con la ejecución del programa:</p> <ul style="list-style-type: none"> <li>• Nombre del programa que ejecuta</li> <li>• Fecha de inicio de la ejecución</li> <li>• Versión de VBS</li> <li>• Fichero Dimensión</li> <li>• Fichero log</li> <li>• Fichero procedure</li> <li>• Fichero validacion</li> <li>• Nombre de la base de datos</li> <li>• Fecha de fin de ejecución</li> </ul>		
<b>Necesidad</b>	Alta/ <b>Media</b> /Baja	<b>Riesgo</b>	Alto/Medio/ <b>Bajo</b>
<b>Prioridad</b>	Alta/ <b>Media</b> /Baja	<b>Estabilidad</b>	Alta/ <b>Media</b> /Baja

Tabla 34: Requisito RF-33

<b>ID</b>	RF-34		
<b>Nombre</b>	Inserción de la base de datos en el hito 3.		
<b>Descripción</b>	Se podrá introducir al fichero de procedure el nombre de la base de datos a utilizar.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 35: Requisito RF-34

<b>ID</b>	RF-35		
<b>Nombre</b>	Leer Fichero formula		
<b>Descripción</b>	Leerá el fichero aserción en <b>texto plano</b> .		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 36: Requisito RF-35

<b>ID</b>	RF-36		
<b>Nombre</b>	Crear Fichero procedure.		
<b>Descripción</b>	Crearé el fichero con la formula en formato <b>SQL</b> .		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 37: Requisito RF-36

<b>ID</b>	RF-37		
<b>Nombre</b>	Crear Fichero validation.		
<b>Descripción</b>	Crearé el fichero validación en formato <b>SQL</b> .		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 38: Requisito RF-37

<b>ID</b>	RF-38		
<b>Nombre</b>	Crear Fichero Log en el hito 3.		
<b>Descripción</b>	Crearé el fichero log.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 39: Requisito RF-38

<b>ID</b>	RF-39		
<b>Nombre</b>	Tratamiento fórmulas		
<b>Descripción</b>	El programa solución en SQL, deberá tratar las formulas por su tipo.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 40: Requisito RF-39

<b>ID</b>	RF-40		
<b>Nombre</b>	Tipo de fórmulas		
<b>Descripción</b>	El tipo de formulas dependerá de el tratamiento de contextos, dimensiones y atributos de dimensión.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 41: Requisito RF-40

<b>ID</b>	RF-41		
<b>Nombre</b>	Variables de la fórmula		
<b>Descripción</b>	Serán tratadas guardando nombre de la variable, nombre concepto básico y valor en caso de null.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 42: Requisito RF-41

<b>ID</b>	RF-42		
<b>Nombre</b>	Precondición		
<b>Descripción</b>	Si existe deberá ser cumplida para poder validar la fórmula.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 43: Requisito RF-42

<b>ID</b>	RF-43		
<b>Nombre</b>	Mensaje de Validación		
<b>Descripción</b>	El programa solución SQL, deberá mostrar un mensaje solución de si se ha pasado la validación.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 44: Requisito RF-43

<b>ID</b>	RF-44		
<b>Nombre</b>	Tratamiento de Filtros.		
<b>Descripción</b>	Si existen filtros en la aserción, la solución SQL se repetirá para cada uno de estos filtros.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 45: Requisito RF-44

#### 4.1.2 Requisitos no funcionales

<b>ID</b>	RNF-01		
<b>Nombre</b>	Estandar de mercado		
<b>Descripción</b>	Cumplir con el estándar SQL:2003 para los ficheros de salida SQL		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 46: Requisito RNF-01

<b>ID</b>	RNF-02		
<b>Nombre</b>	Compatibilidad de Sistema Operativo		
<b>Descripción</b>	Ser compatible con cualquier versión de Windows XP o superior.		
<b>Necesidad</b>	Alta/Media/Baja	<b>Riesgo</b>	Alto/Medio/Bajo
<b>Prioridad</b>	Alta/Media/Baja	<b>Estabilidad</b>	Alta/Media/Baja

Tabla 47 Requisito RNF-02

## 4.2 CASOS DE USO

### 4.2.1 Especificación de los casos de uso

El formato a seguir en la especificación de los casos de uso es el siguiente:

<b>ID</b>	CU-XX	<b>NOMBRE</b>	Nombre del caso de uso.
<b>REQUISITOS ASOCIADOS</b>	Requisitos asociados al caso de uso.		
<b>ACTORES</b>	Actores que pueden realizar el caso de uso.		
<b>OBJETIVO</b>	Objetivo del caso de uso		
<b>PRECONDICIONES</b>	Las condiciones que deben satisfacerse para la realización del caso de uso.		
<b>POSTCONDICIONES</b>	Todo aquello que ocurre como consecuencia de realizar el caso de uso		
<b>ESCENARIO</b>	<ol style="list-style-type: none"> <li>1. Primer paso.</li> <li>2. Segundo paso.</li> <li>3. Etc.</li> </ol>		
<b>CONDICIONES DE FALLO</b>	Posibles errores que pueden surgir durante la realización de un caso de uso, y las acciones que se llevan a cabo.		

Tabla 48: Plantilla Casos de Uso

<b>ID</b>	CU-01	<b>NOMBRE</b>	Ejecución Hito 1
<b>REQUISITOS ASOCIADOS</b>	RF-01, RF-02, RF-03, RF-04		
<b>ACTORES</b>	Usuario		
<b>OBJETIVO</b>	Ejecución del programa xbrl001.vbs para la generación de la inserciones.		
<b>PRECONDICIONES</b>	-		
<b>POSTCONDICIONES</b>	Generación del archivo de inserciones del hito 1.		
<b>ESCENARIO</b>	<ol style="list-style-type: none"> <li>1. Elegir fichero dimensiones</li> <li>2. Elegir nombre fichero log</li> <li>3. Elegir nombre fichero de inserciones.</li> <li>4. Elegir nombre de la base de datos</li> <li>5. Ejecución de xbrl001.vbs</li> <li>6. Generación fichero inserciones</li> </ol>		
<b>CONDICIONES DE FALLO</b>	En caso de Error en fichero de dimensión, no se escribirán las inserciones en el fichero de inserciones.		

Tabla 49: Caso de uso CU-01

<b>ID</b>	CU-02	<b>NOMBRE</b>	Ejecución Hito 2
<b>REQUISITOS ASOCIADOS</b>	RF-14, RF-15, RF-16, RF-17		
<b>ACTORES</b>	Usuario		
<b>OBJETIVO</b>	Ejecución del programa xbrl002.vbs para la generación de las inserciones de contexto, hecho y cabecera.		
<b>PRECONDICIONES</b>	-		
<b>POSTCONDICIONES</b>	Generación del archivo de inserciones del hito 2.		
<b>ESCENARIO</b>	<ol style="list-style-type: none"> <li>1. Elegir fichero instancia</li> <li>2. Elegir nombre fichero log</li> <li>3. Elegir nombre fichero de inserciones.</li> <li>4. Elegir nombre de la base de datos</li> <li>5. Ejecución de xbrl002.vbs</li> <li>6. Generación fichero inserciones</li> </ol>		
<b>CONDICIONES DE FALLO</b>	En caso de Error en fichero instancia, no se escribirán las inserciones en el fichero de inserciones.		

Tabla 50: Caso de uso CU-02



<b>ID</b>	CU-03	<b>NOMBRE</b>	Ejecución Hito 3
<b>REQUISITOS ASOCIADOS</b>	RF-28, RF-29, RF-30, RF-31, RF-32		
<b>ACTORES</b>	Usuario		
<b>OBJETIVO</b>	Ejecución del programa xbrl003.vbs para la generación del código SQL de validación de la taxonomía.		
<b>PRECONDICIONES</b>	-		
<b>POSTCONDICIONES</b>	Generación del archivo SQL de validación de la taxonomía.		
<b>ESCENARIO</b>	<ol style="list-style-type: none"> <li>1. Elegir fichero de aserción</li> <li>2. Elegir nombre fichero log</li> <li>3. Elegir nombre fichero de validación.</li> <li>4. Elegir nombre fichero procedure.</li> <li>5. Elegir nombre de la base de datos</li> <li>6. Ejecución de xbrl003.vbs</li> <li>7. Generación fichero SQL de validacion</li> </ol>		
<b>CONDICIONES DE FALLO</b>	En caso de Error en fichero de aserción, no se generará correctamente el fichero SQL de validación.		

Tabla 51: Caso de uso CU-03

#### 4.2.2 Diagrama de casos de uso

En este apartado se muestra el diagrama de los casos de uso descritos en el apartados anterior.

En este diagrama se puede ver una figura en forma de persona, esta figura representa el usuario del sistema. También podemos ver tres óvalos dentro de un rectángulo. En rectángulo expresa el concepto global del proyecto y cada óvalo representa un caso de uso, uno por cada hito del proyecto.

El usuario puede ejecutar tres acciones que son Ejecución del Hito1, Ejecución del Hito2 y Ejecución del Hito3.

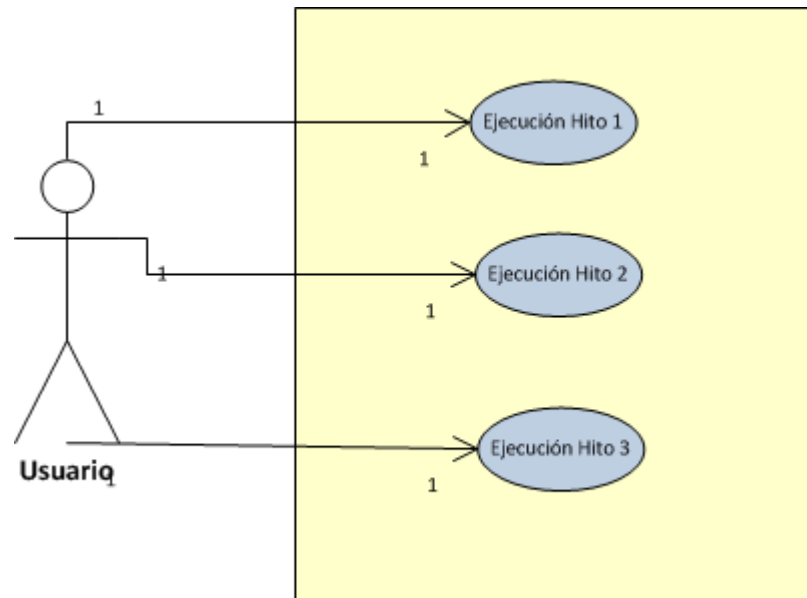


Ilustración 5: Diagrama de casos de uso

### 4.3 IDENTIFICACIÓN DE LOS SUBSISTEMAS

Para la reducción de la complejidad se dividirá el sistema en sistemas más pequeños, de manera que se tratará cada subsistema de manera independiente, lo cual ayudará a la hora manejar el sistema y facilitará los cambios al estar en diferentes subsistemas y también será más fácil la realización de las pruebas al separarlas por subsistema.

Para identificar los subsistemas se puede utilizar diferentes criterios, para la separación en subsistemas hemos elegido la homogeneidad de procesos, la afinidad de requisitos y la separación por hitos. Como se ha comentado anteriormente en diferentes apartados este proyecto se separa en tres hitos relacionados entre sí pero cada uno con su solución de implantación diferente, estos hitos además cumplen que sus requisitos y sus procesos son semejantes. Con lo cual los subsistemas a desarrollar serán Hito1, Hito2 e Hito3.

Subsistemas creados:

- Subsistema Hito 1: Este subsistema consiste en la creación e inserción de todos los metadatos de la taxonomía como las dimensiones, atributos de dimensión, conceptos básicos y relaciones de dimensión-atributo.
- Subsistema Hito 2: Este subsistema consiste en la creación e inserción de los documentos instancia e informes, es decir de los contextos, hechos, cabecera, y relaciones.
- Subsistema Hito 3: Este subsistema consiste en la creación y validación de las aserciones para la validación de los hechos de los documentos instancias XBRL pero en el SGBD.

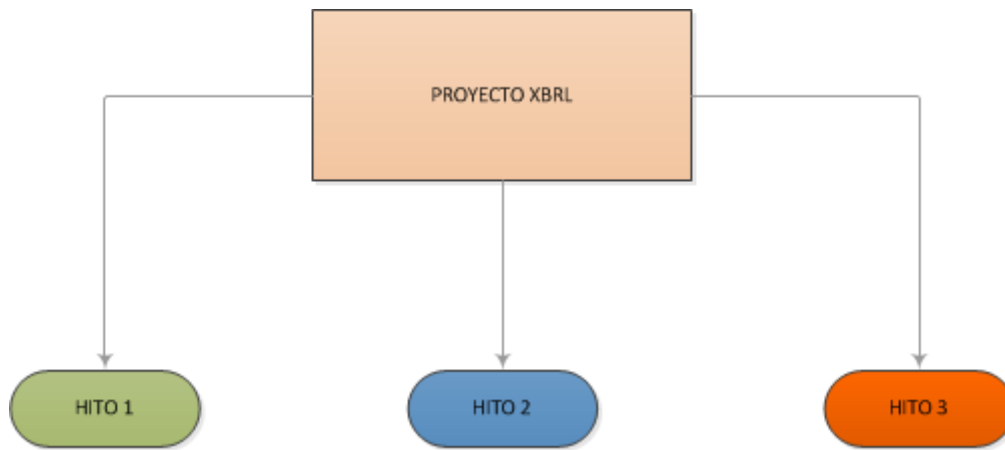


Ilustración 6: Subsistemas de Análisis

Este gráfico de la ilustración 6, muestra la relación entre los diferentes subsistemas. Dentro del sistema general que es este proyecto encontramos el subsistema del Hito 1, el subsistema del Hito 2 y el subsistema del Hito 3.

## 5. DISEÑO TÉCNICO DEL SISTEMA

### 5.1 SELECCIÓN DE LA ARQUITECTURA

En esta apartado comentaremos la selección de la arquitectura que se ha llevado a cabo para la realización de este proyecto y de la tesis doctoral a lo que da apoyo.

En este proyecto se parte de una serie de programas ya desarrollados para la taxonomía FINREP 2008 y parte de FINREP 2012. Por esta razón se tienen unas restricciones en cuanto a lenguajes de programación.

Se ha considerado que los programas se realizan con Visual Basic Script ya que es un lenguaje fácil de entender, y este trabajo está orientado a profesionales con un perfil no especializado en Ingeniería Informática.

El motor de base de datos elegido es SQL Server. Se ha elegido este motor de base de datos porque todos los reguladores usan SQL Server.

Las restricciones en cuando a tiempo con las que contamos son 300 horas, que son las estipulas en la planificación para el desarrollo de este proyecto y para poder finalizar a tiempo. Para la planificación se han tenido en cuenta estos lenguajes.

### 5.2 ARQUITECTURA DE SOPORTE

#### 5.2.1 Identificación de Subsistemas de Diseño

En esta tarea se parte de los subsistemas identificados en los procesos de Análisis. El objetivo es reducir la complejidad y facilitar el mantenimiento, como se comentó en análisis. Cada subsistema posee su propio programa, por lo que cada uno de ellos se puede diseñar independientemente.

La elección de los distintos subsistemas es la misma que la elegida en los subsistemas de análisis que es, Subsistema de Hito 1, subsistema de Hito 2 y subsistema de Hito 3.

#### 5.2.2 Especificación del Entorno Tecnológico

En esta tarea se definen los distintos elementos de la infraestructura técnica que dan soporte al sistema de información. Se define el entorno tecnológico del sistema como la tecnología, tanto software como hardware, que va a requerir para su funcionamiento.

##### 5.2.2.1 Hardware

La solución al problema implicará que la máquina sobre la que se ejecute la solución tenga unas características mínimas. La solución obtenida no requiere una máquina con características muy altas, ya que al querer que la solución la pudiera ejecutar un usuario en su propia máquina se buscó una manera de hacer esta solución fácil y con pocos requisitos. Por lo tanto, se necesitaran las siguientes características para que los programas tengan un buen rendimiento.

- Procesador Intel o AMD con frecuencia mínima de 800 MHz.
- 256 MB de Memoria RAM
- 60 GB de espacio libre en disco duro

### 5.2.2.2 Software

Para ejecutar la aplicación desde un equipo, se recomienda las plataformas:

- Microsoft Windows XP o superior.
- SQL Server 2008 o superior.

### 5.2.3 Diseño de Subsistemas de Soporte

A continuación se va a realizar el diseño de los subsistemas de soporte junto con una descripción de cada uno.

Las tablas que se van a utilizar para el diseño de los subsistemas es el siguiente:

Programa	
<b>Nombre</b>	Nombre descriptivo del programa
<b>Descripción</b>	Descripción completa del programa del subsistema
<b>Subrutinas</b>	Rutinas del programa
<b>Datos</b>	Los datos que posee el programa
<b>Entrada</b>	Parámetros de entrada al programa
<b>Salida</b>	Salida obtenida del programa

Tabla 52: Plantilla Programa

Subrutinas	
<b>Nombre</b>	Nombre descriptivo de la rutina del programa del subsistema
<b>Programa</b>	Programa al que pertenece la rutina.
<b>Descripción</b>	Descripción completa de la rutina del programa del subsistema
<b>Referencias</b>	Referencias de la rutina del programa si realiza alguna interacción con otra rutina del mismo programa.
<b>Proceso</b>	Funcionamiento, paso a paso, de la función. Teniendo en cuenta posibles errores.

Tabla 53: Plantilla Subrutinas

### 5.2.3.1 Subsistema de Hito 1

Programa	
<b>Nombre</b>	Extracción de dimensión y conceptos básicos (xbrl001.vbs)
<b>Descripción</b>	Este programa realiza la extracción del fichero de dimensión de las dimensiones, atributos y conceptos básicos para después crear un fichero de inserciones donde se escriban en formato SQL los insert de estos datos.
<b>Subrutinas</b>	Rutina: main el programa read_dimensions: rutina que lee y crea los datos.
<b>Datos</b>	xbrls002_MiFecha, xbrls002_MiHora xbrls002_Error_Proceso xbrls002_Nombre_Programa xbrls002_arelle_file xbrls002_fso_log xbrls002_fo_log xbrls002_sentenceSQL xbrls002_baseDatos xbrls002_nombre_maquina xbrls002_fichero_log
<b>Entrada</b>	Fichero dimensiones Nombre fichero log Nombre fichero inserciones Nombre base de datos
<b>Salida</b>	Fichero log Fichero inserciones

Tabla 54: Programa xbrl001

Subrutina	
<b>Nombre</b>	Rutina main que ejecuta la acción (Rutina)
<b>Programa</b>	xbrl001.vbs
<b>Descripción</b>	Rutina que lee la entrada al sistema, crea el fichero log y llama a la subrutina read_dimensions.
<b>Referencias</b>	read_dimensions
<b>Proceso</b>	Lee y guarda los datos de entrada al programa.  Crea e inserta los datos para el fichero log.  Llama a read_dimensions

Tabla 55: Subrutina H1: Rutina

Subrutina	
<b>Nombre</b>	Rutina que lee las dimensiones (read_dimensions)
<b>Programa</b>	xbrl001.vbs
<b>Descripción</b>	Rutina que lee el fichero de dimensión y escribe los datos en el fichero de inserciones SQL.
<b>Referencias</b>	Rutina
<b>Proceso</b>	Lee línea a línea el fichero de dimensiones.  Determina que dato se trata en esa línea.  Crea la inserción cuando corresponda del dato en el fichero de inserciones.

Tabla 56: Subrutina H1: read\_dimensions

### 5.2.3.2 Subsistema de Hito 2

Programa	
<b>Nombre</b>	Extracción de contextos y hechos (xbrl002.vbs)
<b>Descripción</b>	Este programa realiza la extracción del fichero instancia de los contextos y los hechos para después crear un fichero de inserciones donde se escriban en formato SQL los insert de estos datos.
<b>Subrutinas</b>	Rutina :main del programa read_documentInstance: subrutina que lee y crea los datos
<b>Datos</b>	xbrls002_MiFecha, xbrls002_MiHora xbrls002_Error_Proceso xbrls002_Nombre_Programa xbrls002_documentInstance xbrls002_fso_log, xbrls002_fo_log, xbrls002_sentenceSQL xbrls002_baseDatos xbrls002_nombre_maquina xbrls002_fichero_log
<b>Entrada</b>	Fichero instancia Nombre fichero log Nombre fichero inserciones Nombre base de datos
<b>Salida</b>	Fichero log Fichero inserciones

Tabla 57: Programa xbrl002



Subrutina	
<b>Nombre</b>	Rutina main que ejecuta la acción (Rutina)
<b>Programa</b>	xbrl002.vbs
<b>Descripción</b>	Rutina que lee la entrada al sistema, crea el fichero log y llama a la subrutina read_ documentInstance.
<b>Referencias</b>	read_ documentInstance
<b>Proceso</b>	Lee y guarda los datos de entrada al programa.  Crea e inserta los datos para el fichero log.  Llama a read_ documentInstance

Tabla 58:Subrutina H2: Rutina

Subrutinas	
<b>Nombre</b>	Rutina que lee el documento instancia(read_ documentInstance)
<b>Programa</b>	xbrl002.vbs
<b>Descripción</b>	Rutina que lee el fichero instacia y escribe los datos en el fichero de inserciones SQL
<b>Referencias</b>	Rutina
<b>Proceso</b>	Lee línea a línea el fichero instacia  Determina que dato se encuentra en esa línea  Crea la inserciones del dato cuando corresponda en el fichero de inserciones.

Tabla 59: Subrutina H2: read\_ documentInstance

### 5.2.3.3 Subsistema de Hito 3

Programa		
<b>Nombre</b>	Validación de taxonomías (xbrl003.vbs)	
<b>Descripción</b>	Este programa realiza la extracción del fichero de aserciones obtenido de Arelle para convertirlo en un fichero SQL de validación de las taxonomías.	
<b>Subrutinas</b>	Rutina :main del programa  spExFiles  read_Formulas  initializeVariables	replacePrecondition(PreconditionVar)  dealVariable (recordFile, nameOfVariable)  dimVariable (str1, dimAttr, dimensionV)

	<p>assertionNameGet(recordFile, assertionName, assertionStr)</p> <p>assertionAnalyze(assertionName, assertionStr)</p> <p>readParameter(recordFile)</p> <p>readExplicitFilters(recordFile, explFilter_Dim, explFilter_AttrDim, conceptExplicitFilter, DimensionExplicitFilter, numExplFilter)</p> <p>replaceString(assertionSentence, assertionSentenceNew, specialFormula_True)</p> <p>variableHeader(recordFile, nameOfVariable)</p> <p>dealPreconditionVariable(recordFile)</p> <p>evaluatePrecondition(PreconditionVar)</p>	<p>mapping</p> <p>isTypeI(assertionTypeI)</p> <p>spTypeI(assertionName_M)</p> <p>spTypeII(assertionName_M)</p> <p>spTypeIII(assertionName_M)</p> <p>spTypeIV(assertionName_M)</p> <p>spTypeV(assertionName_M)</p> <p>spTypeVI(assertionName_M)</p> <p>spTypeVII(assertionName_M)</p> <p>replaceFunctMath(str1)</p> <p>errorMessage(errorNum, comment1, comment2, comment3, comment4, comment5)</p>
<b>Datos</b>	<p>xbrls033_Lab6_MiFecha</p> <p>xbrls033_Lab6_MiHora</p> <p>xbrls033_Lab6_Error_Proceso</p> <p>xbrls033_Lab6_Nombre_Programa</p> <p>xbrls033_Lab6_arelle_file</p> <p>xbrls033_Lab6_fso_log</p> <p>xbrls033_Lab6_fo_log</p> <p>xbrls033_Lab6_SP</p> <p>xbrls033_Lab6_Validation</p> <p>xbrls033_Lab6_baseDatos</p> <p>xbrls033_Lab6_nombre_maquina</p> <p>xbrls033_Lab6_fichero_log</p> <p>fsi1</p> <p>fi1</p> <p>mivar</p> <p>objArgs</p>	<p>newFactVariable</p> <p>assertionDef</p> <p>numConcepts</p> <p>numDim</p> <p>valuePreconditionV</p> <p>DimensionExplicitFilterV</p> <p>fallbackValueV</p> <p>existFallbackValueV</p> <p>numParameters</p> <p>nameParameter</p> <p>nameInFormula</p> <p>valueParameter</p> <p>assertionName</p> <p>assertionSentenceNew</p> <p>specialFormula_True</p> <p>assertionSentence_True</p>

	comment1	xbrls033_Lab6_assertionSet
	comment2	xbrls033_Lab6_valueAssertion
	comment3	xbrls033_Lab6_conceptName
	comment4	xbrls033_Lab6_factVariable
	comment5	xbrls033_Lab6_explicitDimension
	errorNum	xbrls033_Lab6_parameter
	xbrls033_Lab6_fo_SP	xbrls033_Lab6_comma
	xbrls033_Lab6_fso_SP	xbrls033_Lab6_dollar
	xbrls033_Lab6_fo_Validation	xbrls033_Lab6_equal
	xbrls033_Lab6_fso_Validation	xbrls033_Lab6_SearchChar
	variablePrecondicion	xbrls033_Lab6_Dim_
	explFilter_Dim	xbrls033_Lab6_quotation_marks
	explFilter_AttrDim	xbrls033_Lab6_blank
	onceptExplicitFilter	xbrls033_Lab6_idiv
	DimensionExplicitFilde	xbrls033_Lab6_false
	numExplFilter	xbrls033_Lab6_conceptDataType
	variableNames	xbrls033_Lab6_precondition
	variableNumber	xbrls033_Lab6_every
	conceptExplicitFilterV	xbrls033_Lab6_i
	numExplFilterV	xbrls033_Lab6_in
	variableID	xbrls033_Lab6_satisfies
	explFilterV_Dim	xbrls033_Lab6_formula
	explFilterV_AttrDim	xbrls033_Lab6_or
	newAssertionGroup	xbrls033_Lab6_and
	newAssertion	
<b>Entrada</b>	Fichero aserciones Nombre fichero log Nombre fichero procedure Nombre fichero validate	

	Nombre base de datos
<b>Salida</b>	Fichero log Fichero procedure SQL Fichero Validate SQL

Tabla 60: Programa xbrl003

Subrutina	
<b>Nombre</b>	Rutina main que ejecuta la acción (Rutina)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Rutina que lee la entrada al sistema, crea el fichero log y llama a la subrutina spExFiles y a read_Formulas.
<b>Referencias</b>	spExFiles read_Formulas
<b>Proceso</b>	Lee y guarda los datos de entrada al programa.  Crea e inserta los datos para el fichero log.  Llama a spExFiles  Llama a read_Formulas

Tabla 61: Subrutina H3: Rutina

Subrutina	
<b>Nombre</b>	Rutina que crea fichero solución SQL (spExFiles)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que crea introduce a base de datos en los fichero solución
<b>Referencias</b>	Rutina
<b>Proceso</b>	Crea y añade base de datos al fichero valide  Crea y añade base de datos al fichero procedure.

Tabla 62: Subrutina H3: spExFiles

Subrutina		
<b>Nombre</b>	Rutina que lee líneas del fichero de aserciones(read_Formulas)	
<b>Programa</b>	xbrl003.vbs	
<b>Descripción</b>	Subrutina que lee las líneas del fichero de aserciones y llama a los métodos según corresponda la información de la línea.	
<b>Referencias</b>	Rutina Mapping initializeVariables assertionNameGet assertionAnalyze	variableHeader readParameter readExplotFilters dealVariable dealPreconditionVariable
<b>Proceso</b>	Lee cada línea del fichero Llama a el método correpondiente en cada lien Si empieza nueva aserción o el fichero se acabó llama a mapping	

Tabla 63: Subrutina H3: read\_formulas

Subrutina	
<b>Nombre</b>	Rutina que inicializa las variables (initializeVariables)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que inicializa las variables del programa.
<b>Referencias</b>	read_Formulas
<b>Proceso</b>	Inicializa las variables del programa.

Tabla 64: Subrutina H3: initializeVariables

Subrutina	
<b>Nombre</b>	Rutina que recupera el nombre de la aserción (assertionNameGet)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que recupera el nombre de la aserción.
<b>Referencias</b>	read_Formulas
<b>Proceso</b>	Busca y guarda el nombre de la aserción.

Tabla 65: Subrutina H3: assertionNameGet

Subrutina	
<b>Nombre</b>	Rutina que recupera la fórmula de la aserción (assertionAnalyze)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que recupera la fórmula de la aserción
<b>Referencias</b>	read_Formulas replaceString
<b>Proceso</b>	Busca la fórmula  Llama a replaceString  Guarda la fórmula

Tabla 66: Subrutina H3: assertionAnalyze

Subrutina	
<b>Nombre</b>	Rutina que reemplaza caracteres (replaceString)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que cambia los caracteres de la fórmula por caracteres SQL.
<b>Referencias</b>	assertionAnalyze
<b>Proceso</b>	Reemplaza caracteres de la fórmula.

Tabla 67: Subrutina H3: replaceString

Subrutina	
<b>Nombre</b>	Rutina que lee los parámetros de la aserción (readParameter)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que recupera el nombre de la aserción.
<b>Referencias</b>	read_Formulas
<b>Proceso</b>	Busca y guarda los parámetros de la asección.

Tabla 68: Subrutina H3: readParameter

Subrutina	
<b>Nombre</b>	Rutina que lee los filtros de la aserción (readExplicitFilters)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que guarda los filtros de la aserción.
<b>Referencias</b>	read_Formulas
<b>Proceso</b>	Busca los conceptos básicos, atributos y dimensión de la aserción y los guarda.

Tabla 69: Subrutina H3: readExplicitFilters

Subrutina	
<b>Nombre</b>	Rutina que lee las variables de la aserción (variableHeader)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que guarda los valores de las variables de la aserción.
<b>Referencias</b>	read_Formulas errormessage
<b>Proceso</b>	Busca el nombre de la variable  Busca el valor si es null de la variable  Guarda valores encontrados.

Tabla 70: Subrutina H3: variableHeader

Subrutina	
<b>Nombre</b>	Rutina que lee la precondition de la aserción (dealPreconditionVariable)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que guarda la precondition.
<b>Referencias</b>	read_Formulas replacePrecondition evaluatePrecondition
<b>Proceso</b>	Busca la precondition  Llama a replacePrecondition  Llama a evaluatePrecondition  Guarda la precondition

Tabla 71: Subrutina H3: dealPreconditionVariable

Subrutina		
<b>Nombre</b>	Rutina de mensaje de error (errormessage)	
<b>Programa</b>	xbrl003.vbs	
<b>Descripción</b>	Subrutina que imprime el error.	
<b>Referencias</b>	variableHeader mapping spTypeI spTypeII	spTypeIII spTypeIV spTypeV
<b>Proceso</b>	Imprime los errores dados en el programa.	

Tabla 72: Subrutina H3: errormessage

Subrutina	
<b>Nombre</b>	Rutina que reemplaza los caracteres de la precondition (replacePrecondition)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que reemplaza los caracteres de la precondition.
<b>Referencias</b>	dealPreconditionVariable
<b>Proceso</b>	Busca los caracteres en la precondition y los transforma a caracteres SQL.

Tabla 73: Subrutina H3: replacePrecondition

Subrutina	
<b>Nombre</b>	Rutina que evalúa la precondition (evaluatePrecondition)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que evalúa la precondition
<b>Referencias</b>	dealPreconditionVariable
<b>Proceso</b>	Determina si la precondition es de AND u OR. Guarda las variables en variables de la precondition.

Tabla 74: Subrutina H3: evaluatePrecondition



Subrutina	
<b>Nombre</b>	Rutina que lee los filtros de las variables(dealVariable)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que guarda los valores de los filtros de las variables.
<b>Referencias</b>	read_Formulas dimVariable
<b>Proceso</b>	Llama a dimVariable si tenemos dimensión Busca los conceptos básicos si los hay Guarda los valores encontrados

Tabla 75: Subrutina H3: dealVariable

Subrutina	
<b>Nombre</b>	Rutina que lee las variables de la aserción (dimVariable)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que guarda los valores de las variables de la aserción.
<b>Referencias</b>	dealVariable
<b>Proceso</b>	Busca los atributos y dimensiones de la variable Guarda valores encontrados.

Tabla 76: Subrutina H3: dimVariable

Subrutina		
<b>Nombre</b>	Rutina que distribuye tipos (mapping)	
<b>Programa</b>	xbrl003.vbs	
<b>Descripción</b>	Subrutina que mira el tipo de aserción y lo manda el método correspondiente.	
<b>Referencias</b>	read_Formulas isTypeI spTypeI spTypeII spTypeIII	spTypeIV spTypeV spTypeVI spTypeVII
<b>Proceso</b>	Llama a isTypeI	

	Manda a cada aserción a la subrutina que le corresponde por cada tipo.
--	------------------------------------------------------------------------

Tabla 77: Subrutina H3: mapping

Subrutina	
<b>Nombre</b>	Rutina que asigna el tipo de aserción (isTypeI)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que asigna tipo de aserción dependiendo de sus datos.
<b>Referencias</b>	mapping
<b>Proceso</b>	Busca los datos de la aserción y con respecto a esto le asigna un tipo u otro.

Tabla 78: Subrutina H3: isTypeI

Subrutina	
<b>Nombre</b>	Rutina para tipo 1 (spTypeI)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que crea el procedure SQL para el tipo 1
<b>Referencias</b>	mapping
<b>Proceso</b>	Genera Validación SQL según datos y forma de la aserción

Tabla 79: Subrutina H3: spTypeI

Subrutina	
<b>Nombre</b>	Rutina para tipo 2 (spTypeII)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que crea el procedure SQL para el tipo 2
<b>Referencias</b>	mapping
<b>Proceso</b>	Genera Validación SQL según datos y forma de la aserción

Tabla 80: Subrutina H3: spTypeII

Subrutina	
<b>Nombre</b>	Rutina para tipo 3 (spTypeIII)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que crea el procedure SQL para el tipo 3
<b>Referencias</b>	mapping
<b>Proceso</b>	Genera Validación SQL según datos y forma de la aserción

Tabla 81: Subrutina H3: spTypeIII

Subrutina	
<b>Nombre</b>	Rutina para tipo 4 (spTypeIV)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que crea el procedure SQL para el tipo 4
<b>Referencias</b>	mapping
<b>Proceso</b>	Genera Validación SQL según datos y forma de la aserción

Tabla 82: Subrutina H3: sptypeIV

Subrutina	
<b>Nombre</b>	Rutina para tipo 5 (spTypeV)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que crea el procedure SQL para el tipo 5
<b>Referencias</b>	mapping
<b>Proceso</b>	Genera Validación SQL según datos y forma de la aserción

Tabla 83: Subrutina H3: sptypeV

Subrutina	
<b>Nombre</b>	Rutina para tipo 6 (spTypeVI)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que crea el procedure SQL para el tipo 6
<b>Referencias</b>	mapping
<b>Proceso</b>	<p>Crea cursos</p> <p>Busca valor variables</p> <p>Comprueba precondition</p> <p>Valida fórmula</p>

Tabla 84: Subrutina H3: sptypeVI

Subrutina	
<b>Nombre</b>	Rutina para tipo 7 (spTypeVII)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que crea el procedure SQL para el tipo 7
<b>Referencias</b>	mapping
<b>Proceso</b>	<p>Crea filtros de la aserción</p> <p>Crea variables uniendo filtros y valores de la variable</p> <p>Valida fórmula.</p>

Tabla 85: Subrutina H3: sptypeVII

Subrutina	
<b>Nombre</b>	Rutina que reemplaza caracteres (replaceFunctMath)
<b>Programa</b>	xbrl003.vbs
<b>Descripción</b>	Subrutina que reemplaza caracteres
<b>Referencias</b>	spTypeII
<b>Proceso</b>	Reemplaza "idiv" por "/"

Tabla 86: Subrutina H3: replaceFunctMath

#### 5.2.4 Diagrama de flujo

### 5.2.4.1 Subsistema de Hito

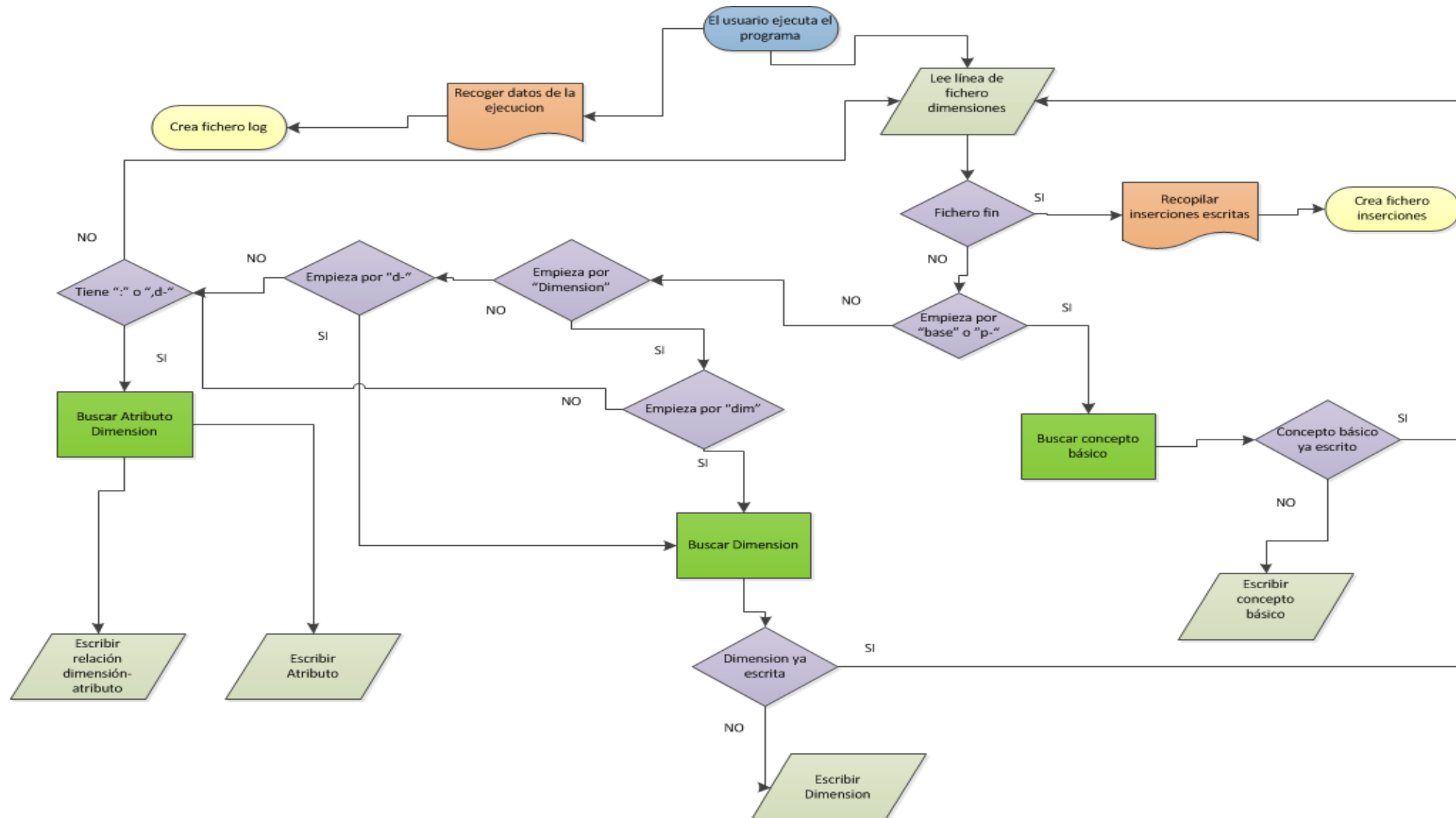


Ilustración 7: Diagrama de Flujo Hito 2

### 5.2.4.2 Subsistema de Hito 2

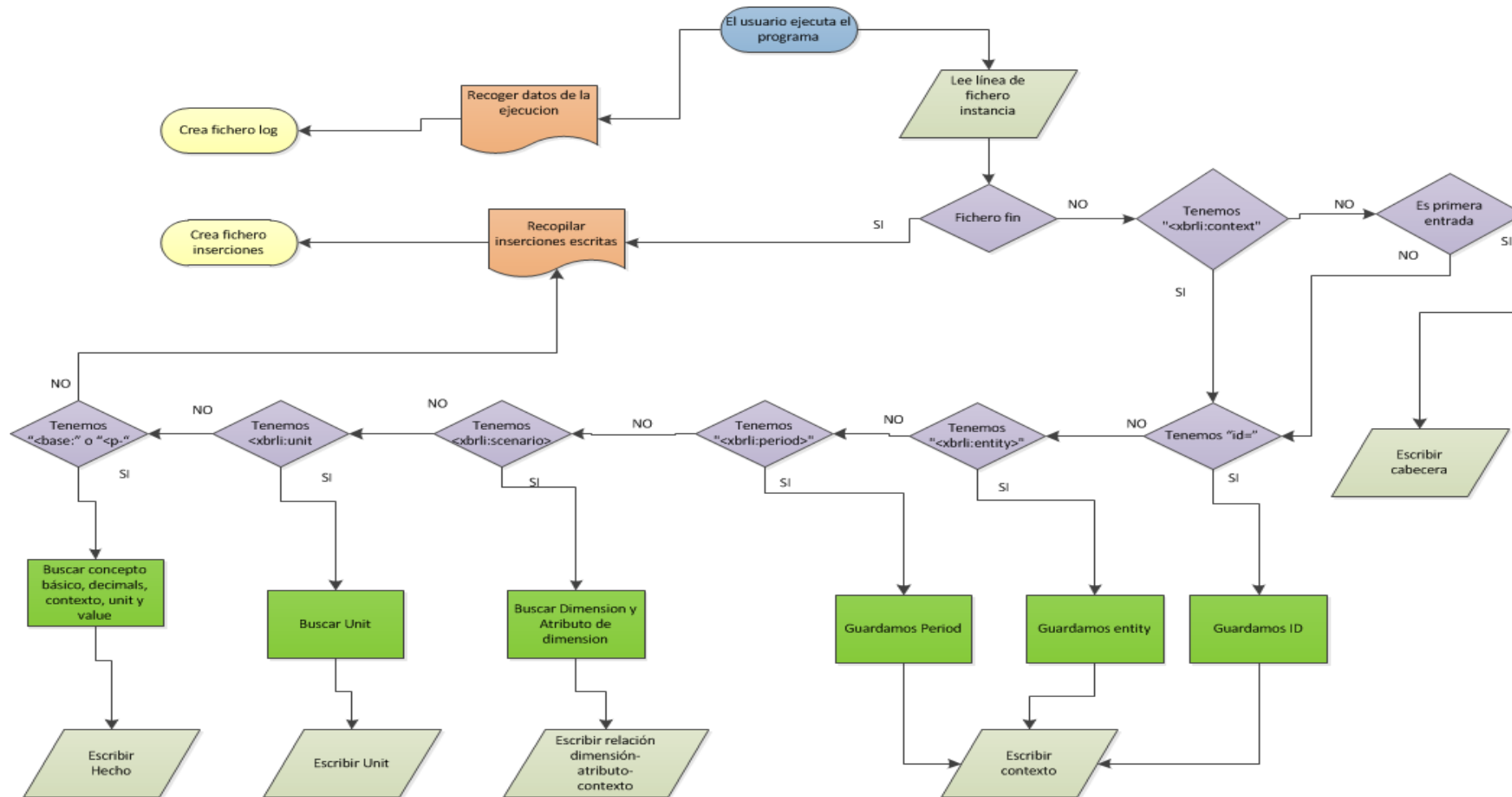


Ilustración 8: Diagrama de Flujo Hito 2

#### **5.2.4.3 Subsistema de Hito 3**

Para el desarrollo de este subsistema se ha basado en tipos de aserciones, todas las aserciones tendrán un tratamiento común de sus datos pero cada una de ella pertenecerá a un tipo. Se mostrará el diagrama de flujo del tratamiento común de aserciones modificado para las taxonomías de este proyecto y los diagramas de los tipos 6 y 7 que se desarrollaran en este proyecto.

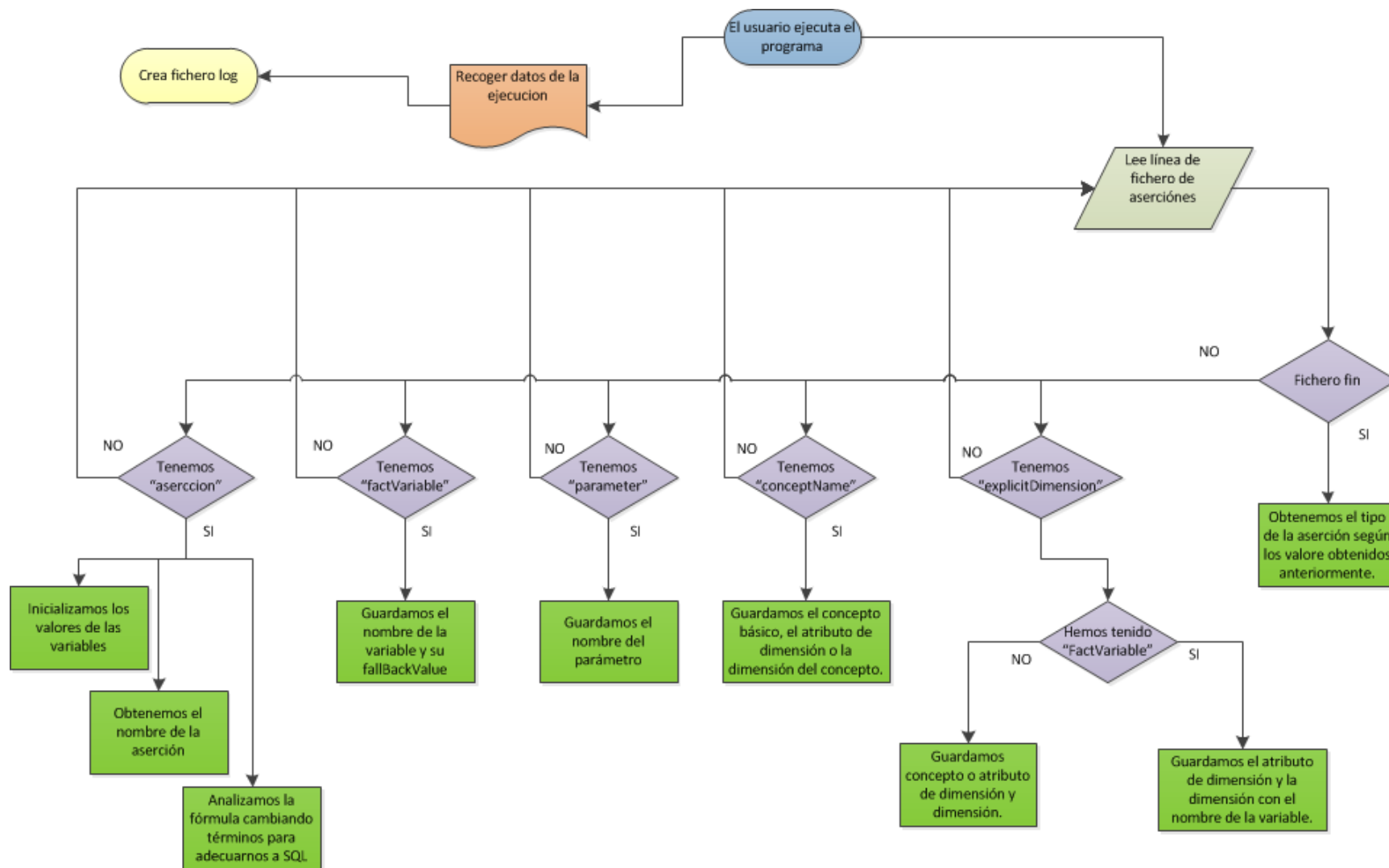


Ilustración 9:Diagrama de Flujo H3 general



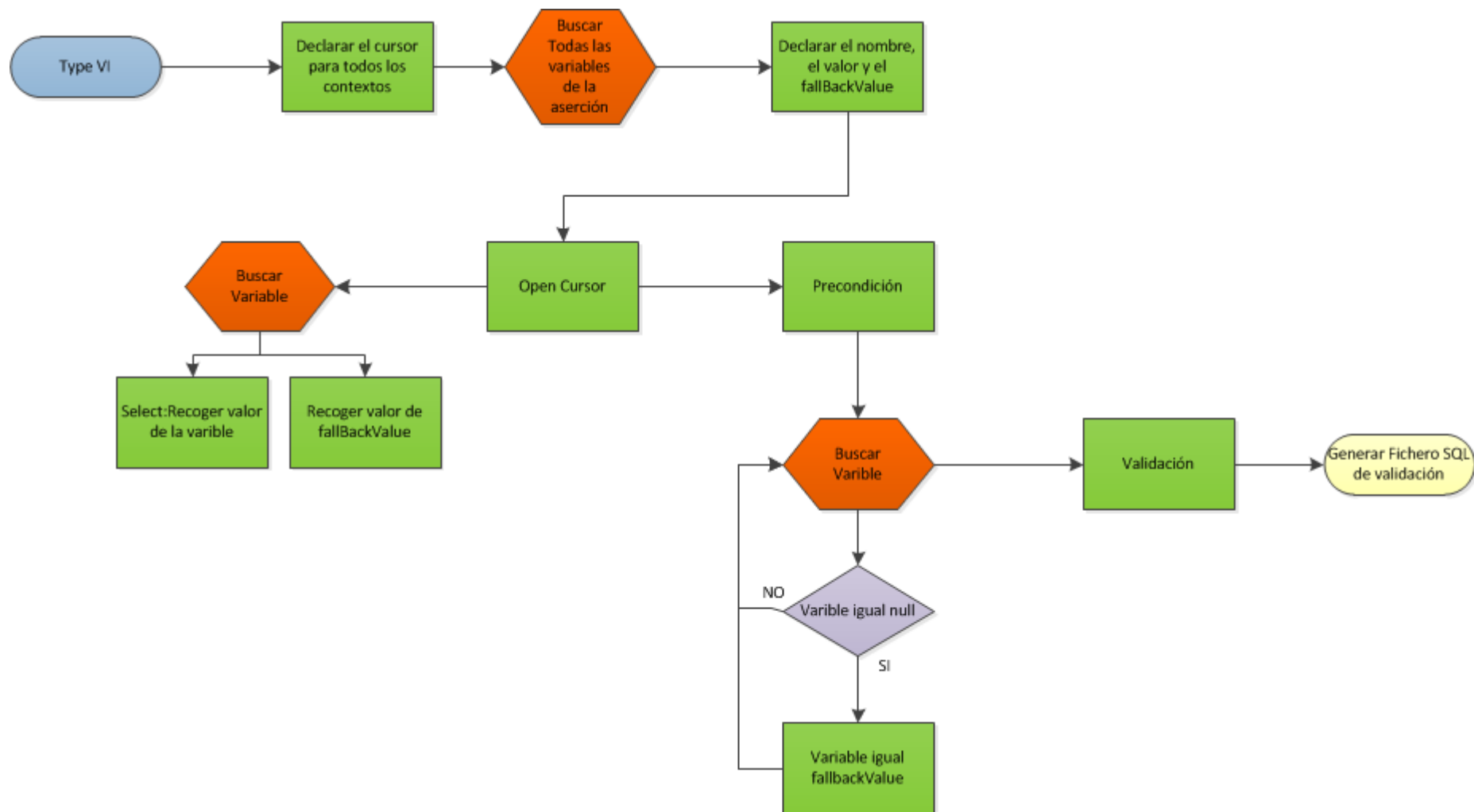


Ilustración 10: Diagrama de Flujo H3 tipo 6

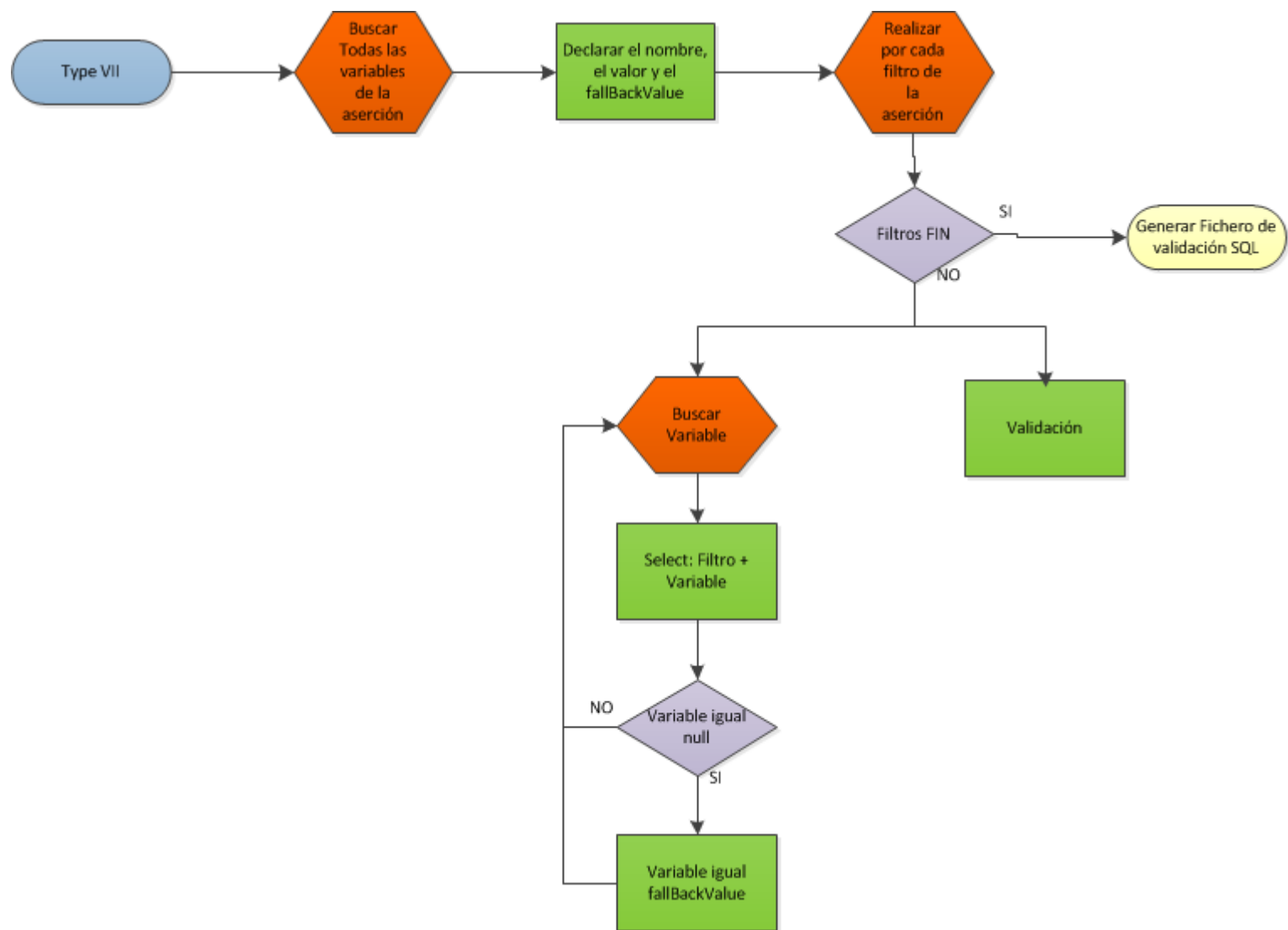


Ilustración 11: Diagrama de Flujo H3 tipo 7

### 5.2.5 Diseño de casos de uso reales

En esta apartado se estudian los casos de uso reales para el desarrollo de la arquitectura. En este caso los casos de uso no han cambiado y se corresponden con los subsistemas de soporte de la siguiente manera:

<b>CASO DE USO</b>	CU-01 Ejecución Hito 1
<b>SUBSISTEMA DE SOPORTE</b>	Subsistema de soporte de Hito 1

Tabla 87: Relación subsistema H1

<b>CASO DE USO</b>	CU-02 Ejecución Hito 2
<b>SUBSISTEMA DE SOPORTE</b>	Subsistema de soporte de Hito 2

Tabla 88: Relación subsistema H2

<b>CASO DE USO</b>	CU-03 Ejecución Hito 3
<b>SUBSISTEMA DE SOPORTE</b>	Subsistema de soporte de Hito 3

Tabla 89: Relación subsistema H3

## 5.3 DISEÑO FÍSICO DE DATOS

### 5.3.1 Diseño del Modelo Físico de Datos

La base de datos a utilizar para la inserción y la transformación de las taxonomías XBRL, será una base de datos relacional, ya que como se ha comentado este proyecto se realiza en el gestor de base de datos de Microsoft SQL Server.

El modelo de base de datos ya viene predefinido en este proyecto y es el mismo que en el apartado de situación actual. Se muestra aquí también el modelo de base de datos para comodidad del lector.

UML Data Model of a Document Instance FINREP 2012.  
(XBRL proof of concept.  
Release, 2011-06-12).

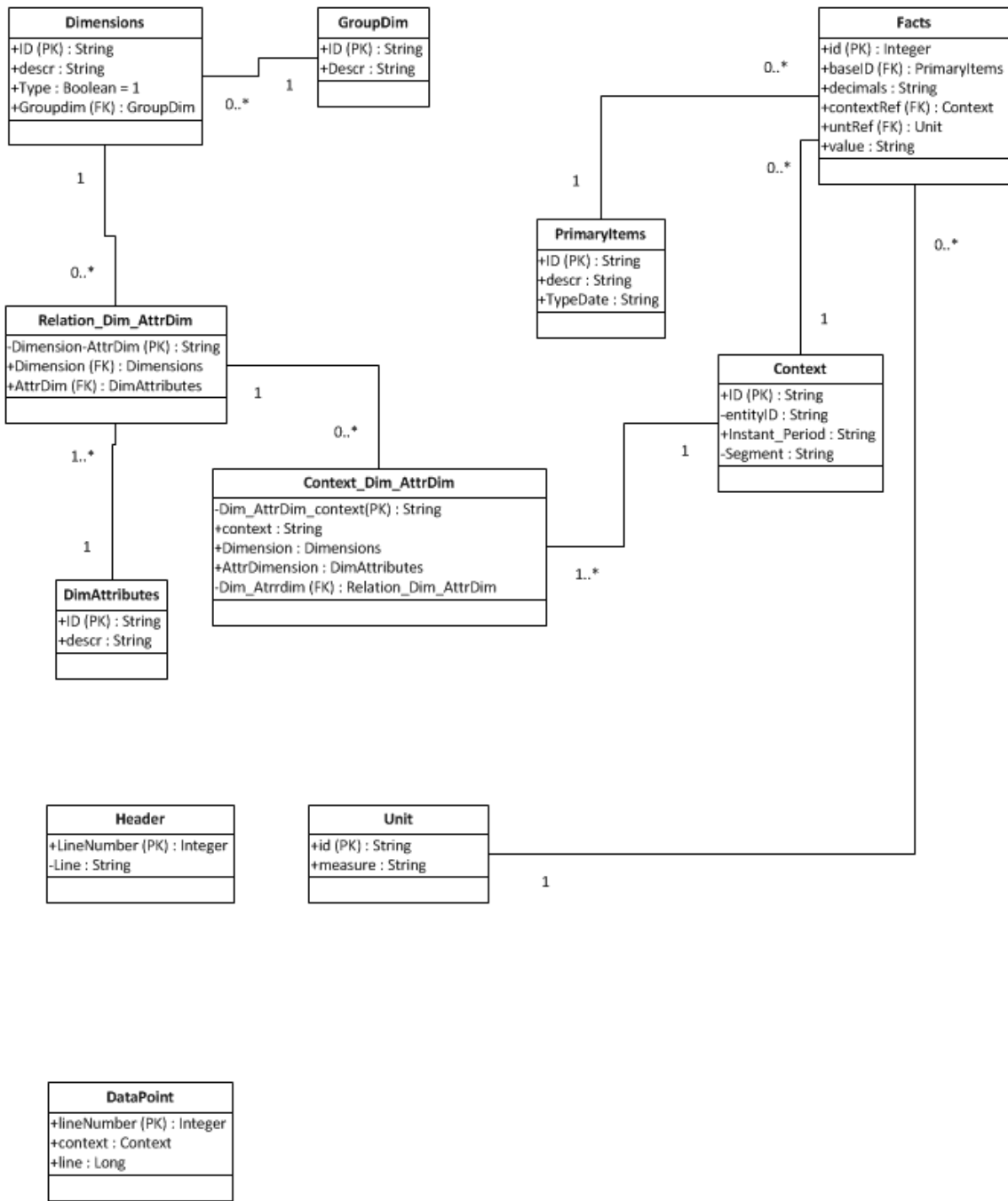


Tabla 90: Modelo base de datos 2

### 5.3.2 Elaboración de Especificaciones del Modelo Físico de Datos

En este apartado se muestra el Script SQL de generación del modelo de datos:

```
use TXBRL
IF OBJECT_ID(N'Facts', N'U') IS NOT NULL
DROP TABLE Facts;
```

```
go
IF OBJECT_ID(N'Context_Dim_AttrDim', N'U') IS NOT NULL
DROP TABLE Context_Dim_AttrDim;
go
IF OBJECT_ID(N'PrimaryItems', N'U') IS NOT NULL
DROP TABLE PrimaryItems;
go

IF OBJECT_ID(N'Relation_Dim_AttrDim', N'U') IS NOT NULL
DROP TABLE Relation_Dim_AttrDim;
go

IF OBJECT_ID(N'Dimensions', N'U') IS NOT NULL
DROP TABLE Dimensions;
go

IF OBJECT_ID(N'DimAttributes', N'U') IS NOT NULL
DROP TABLE DimAttributes;
go

IF OBJECT_ID(N'Context', N'U') IS NOT NULL
DROP TABLE Context;
go

IF OBJECT_ID(N'Header', N'U') IS NOT NULL
DROP TABLE Header;
go

IF OBJECT_ID(N'Unit', N'U') IS NOT NULL
DROP TABLE Unit;
go

create table PrimaryItems (
    ID varchar(100) primary key,
    descr varchar(60),
    Tipo_Date varchar(20));
go

create table Dimensions (
    ID varchar(100) primary key,
    descr varchar(60),
    Type bit default(1));
go

create table DimAttributes (
    ID varchar(100) primary key,
    descr varchar(60))
go

create table Relation_Dim_AttrDim (
    Dimension varchar(100) not null references Dimensions(ID),
    Attrdim varchar(100) not null references DimAttributes(ID),
    Primary key (Dimension, Attrdim));
go

create table Context (
    ID nvarchar(100) primary key,
    entityID nvarchar(100) not null,
    instantPeriod varchar(20),
    segment nvarchar(200) null);
go

create table Context_Dim_AttrDim (
    contextID nvarchar(100) not null references Context(ID),
    dimension varchar(100) not null,
    attrDim varchar(100) not null,
    constraint PK_Context_Dim_AttrDim primary key (dimension, attrDim, contextID),
    constraint FK_Relation_Dim_AttrDim foreign key (dimension, attrDim)
```

```

references Relation_Dim_AttrDim (Dimension, Attrdim))
go

create table Header (
    lineNumber int identity(1,1) primary key,
    line ntext)
go

create table Unit (
    id varchar(100) primary key,
    measure varchar(30) not null)
go

create table Facts (
    id int identity(1,1) primary key,
    baseID varchar(100) not null constraint fk_PrimaryItems references PrimaryItems(ID),
    decimals varchar(50) null,
    contextRef nvarchar(100) not null constraint fk_Context references Context(ID),
    unitRef varchar(100) null constraint fk_Unit references Unit(ID),
    value varchar(200) not null)
go

IF OBJECT_ID(N'dataPoint', N'U') IS NOT NULL
DROP TABLE dataPoint;
go

```

Tabla 91: Script Generación de Tablas

## 5.4 IMPLEMENTACIÓN DE LA SOLUCIÓN

### 5.4.1 Hito 1

#### 5.4.1.1 Parte 1: FINREP 2012

Como se ha explicado brevemente en el estudio y alcance del hito 1, se va a realizar un programa en Visual Basic Script que lea los conceptos y dimensiones obtenidos por Arelle y transforme estos datos en inserciones de la base de datos, con esta parte se obtendrán los **conceptos básicos**, dimensiones, atributos de dimensión y la relación en las dimensiones y sus atributos.

A continuación se muestran dos imágenes del archivo dimensiones2012.txt obtenido pro Arelle.

En la primera podemos ver lo que se llaman **PrimaryItems** (o conceptos básicos), que empiezan por *base* como pueden ser **ad1** o **mi9**. Y en la segunda podemos ver las dimensiones y los atributos de dimensiones. Para definir esta parte, se empieza por *dimensión* y lo que sigue por *dim*, serían las dimensiones como puede ser **AL** y **EQ** y lo siguiente serán los atributos de dimensión como puede ser **dCT:x1** o **dCT:x44**

```
Base items: Balance sheet items,,,,,,,,
,base:ad1,,,,,,,,
,,base:mi2,,,,,,,,domain-member,,,true
,,,base:mi1,,,,,,,,domain-member,,,true
,,,base:mi13,,,,,,,,domain-member,,,true
,,,base:mi9,,,,,,,,domain-member,,,true
,,,base:mi8,,,,,,,,domain-member,,,true
,,,base:md3,,,,,,,,domain-member,,,true
,,,base:mi13,,,,,,,,domain-member,,,true
Base items: Other items,,,,,,,,
,base:ad1,,,,,,,,
,,base:sd6,,,,,,,,domain-member,,,true
```

Ilustración 12: FINREP 2012 Conceptos básicos Arelle

```
Dimension: Categories of assets and liabilities,,,,,,,,
,dim:AL,,,,,,,,
,,dCT:x1,,,,,,,,dimension-domain,,,true
,,,dCT:x12,,,,,,,,domain-member,,,true
,,,dCT:x4,,,,,,,,domain-member,,,true
,,,dCT:x7,,,,,,,,domain-member,,,true
,,,dCT:x22,,,,,,,,domain-member,,,true
,,,dCT:x23,,,,,,,,domain-member,,,true
,,,dCT:x13,,,,,,,,domain-member,,,true
,,,dCT:x38,,,,,,,,domain-member,,,true
,,,dCT:x9,,,,,,,,domain-member,,,true
,,,dCT:x28,,,,,,,,domain-member,,,true
,,,dCT:x44,,,,,,,,domain-member,,,true
Dimension: Categories of equity,,,,,,,,
,dim:EQ,,,,,,,,
,,dCT:x1,,,,,,,,dimension-domain,,,true
```

Ilustración 13: FINREP 2012 Dimensiones Arelle

Analizando estos datos del fichero de dimensiones obtenido de Arelle y de acuerdo con los requisitos del sistema se puede proceder al desarrollo del programa que trate este fichero y obtenga el fichero SQL.

#### 5.4.1.1.1 Desarrollo del programa xbrl001.vbs

En este apartado se va a describir como actual el programa xbrl001.vbs para la creación de las inserciones en FINREP 2012.

1. Entrada de cuatro argumentos por parámetro (fichero Arelle, fichero log, fichero SQL, y nombre base de datos)
2. Leemos cada línea del fichero de la aserción y así vamos procesando de manera individualizada los datos que nos interesan de cada línea.

**SI** empieza por “base:” es un concepto básico

**SI** no insertamos antes este concepto

**Insertar** en base de datos

**SI** empieza por “dimensión:” vamos a procesar una dimensión con sus atributos

**Entonces** isGroupDimensions=1

**SI** empieza por “dim” y no insertamos antes

**Insertar** nombre de dimensión

**SI** ya insertamos nombre de dimensión y isGroupDimensions=1

**Insertar** atributo de dimensión

**Insertar** relación dimensión- atributo de dimensión

#### 5.4.1.2 Parte 2: Estudio de otras Taxonomías

Esta parte del primer hito consiste en el estudio de otras taxonomías, que sistema de ordenación utilizan y si este es compatible con el programa generado anteriormente.

En caso de que el programa `xbrl001.vbs` no sea compatible se estudiarán los cambios necesarios y su coste para que este programa sea compatible.

Las dos taxonomías que estudiaremos serán FINREP 2008 y Solvency II.

##### 5.4.1.2.1 FINREP 2008

Esta taxonomía está dividida en informes cada una con su fichero instancia, para este estudio se tratará el informe 6610.

Siguiendo los mismos pasos que en la parte 1 de este primer hito, para el fichero obtenido de Arelle con esta taxonomía ejecutamos el programa en VBS `xbrl001.vbs` obtenido anteriormente.

De la ejecución del programa `xbrl01.vbs` no se obtiene ningún resultado y se procede a mirar el fichero obtenido de la instancia parado por Arelle.

```
Dimensions Relationships,,,,,,,,Arcrole,CntxElt,Closed,Usable
Balance Consolidado Público,,,,,,,,
,ANEJO III.1 BALANCE CONSOLIDADO PÚBLICO ,,,,,,
,,ACTIVO,,,,,domain-member,,,true
,,,Caja y depósitos en bancos centrales,,,,,domain-member,,,true
,,,Cartera de negociación,,,,,domain-member,,,true
,,,Depósitos en entidades de crédito,,,,,domain-member,,,true
,,,Crédito a la clientela ,,,,domain-member,,,true
,,,Valores representativos de deuda,,,,,domain-member,,,true
,,,Instrumentos de capital,,,,,domain-member,,,true
,,,Derivados de negociación,,,,,domain-member,,,true
,,,Pro-memoria: Prestados o en garantía,,,,,domain-member,,,true
,,,Otros activos financieros a valor razonable con cambios en pérdidas y ganancias,,,,,domain-member,,,true
,,,Depósitos en entidades de crédito,,,,,domain-member,,,true
```

#### Ilustración 14: FINREP 2008 Fichero Arelle

Observando este fichero, podemos resultar que los nombres están escritos de la manera que los verá el usuario y que no siguen el mismo formato que FINREP 2012, con sus atributos *base*, *dimensión* y *dim*, con lo cual para la generación del archivo SQL de inserciones sería necesario un nuevo programa en VBS que procesara este tipo de taxonomías.

El costo de este nuevo programa sería un costo alto, ya que es la creación desde cero. Como la taxonomía de la que estamos intentando ejecutar es una taxonomía antigua, año 2008, y ya existe la nueva versión de 2012, no se considera de utilidad desarrollar un programa nuevo para una taxonomía antigua con un coste elevado.

##### 5.4.1.2.2 Solvency II

Siguiendo el mismo procedimiento que en el estudio de FINREP 2008, obtenemos de la instancia de Solvency II con Arelle el fichero de dimensiones para esta taxonomía.

Ejecutando el programa de VBS `xbrl001.vbs`, no obtenemos ninguna inserción y se procede a investigar el formato del documento de datos de Arelle.

Los conceptos básicos para Solvency II se estructuran de la siguiente forma en la que podemos ver que un concepto básico sería AS1 o A26 por ejemplo.



```
,,p-BS_Cl:AS1,,,domain-member,,,true  
,,p-BS_Cl:AS24A,,,domain-member,,,true  
,,p-BS_Cl:AS24,,,domain-member,,,true  
,,p-BS_Cl:A2,,,domain-member,,,true  
,,p-BS_Cl:AS2,,,domain-member,,,true  
,,p-BS_Cl:A26,,,domain-member,,,true  
,,p-BS_Cl:AS26,,,domain-member,,,true  
,,p-BS_Cl:A3,,,domain-member,,,true
```

#### Ilustración 15: Conceptos Básicos en Solvency2

Las dimensiones y Atributos para Solvency II se estructuran de la siguiente forma en la que podemos ver que la dimensión sería *PeriodicityDimension* y los atributos *Yearly*, *Quarterly* y *AdHoc*.

```
,,d-per:PeriodicityDimension,,,  
,,d-per:PeriodicityDomain,,,dimension-domain,,,true  
,,d-per:Yearly,,,domain-member,,,true  
,,d-per:Quarterly,,,domain-member,,,true  
,,d-per:AdHoc,,,domain-member,,,true
```

#### Ilustración 16: Dimensiones en Solvency2

En esta investigación vemos que los conceptos básicos empiezan por *p-*, lo que tiene sentido gracias a su término en inglés Primary Items, las dimensiones empiezan por *d-*, de dimensión. Esto lo podemos asemejar a *base*, *dimensión* y *dim* en FINREP 2012.

Viendo esta semejanza el coste añadido para que el programa *xbrl001.vbs* trate también las taxonomías de Solvency II es pequeño, con lo cual vamos a modificar este programa de manera que pueda tratar taxonomías de FINREP 2012 y de Solvency II.

##### 5.4.1.2.1 Modificación del programa *xbrl001.vbs* para Solvency II

En este apartado vamos a modificar el programa de VBS para que trate Solvency II, de manera que cuando inserte un concepto básico también lo haga si empieza por *p-* y si se inserta una dimensión también nos inserte cuando sea *d-*.

La funcionalidad del programa es muy semejante y así con esta modificación cumplimos uno de los principales objetivos ideales de este proyecto que es un programa que transforme cualquier taxonomía en XBRL a SQL.

En esta parte se va a describir como actual el programa *xbrl001.vbs* para la creación de las inserciones en Solvency II.

1. Entrada de cuatro argumentos por parámetro (fichero Arelle, fichero log, fichero SQL, y nombre base de datos)
2. Leemos cada línea del fichero de la aserción y así vamos procesando de manera individualizada los datos que nos interesan de cada línea.

**SI** empieza por “p-” es un concepto básico

**SI** no insertamos antes este concepto

**Insertar** en base de datos

**SI** empieza por “d-” vamos a procesar una dimensión con sus atributos

**Entonces** yaDimension=1

**SI** yaDimension=1

**Insertar** nombre de dimensión

**SI** ya insertamos nombre de dimensión y contiene "d-"

**Insertar** atributo de dimensión

**Insertar** relación dimensión- atributo de dimensión

#### 5.4.1.3 Diagrama de flujo

#### 5.4.1.4 Manual para el usuario

Para la ejecución de este programa se utilizará el archivo `xbrl001.cmd`, en el cual se debe poner la ruta de los ficheros y los parámetros del programa sabiendo que: el primero corresponde al fichero de Arelle, el segundo al fichero log, el tercero al fichero de inserciones y el cuarto al nombre de la base de datos.

El resultado será un archivo SQL con las inserciones obtenidas, el cual se podrá ejecutar en la base de datos correspondiente, con las tablas ya creadas y se obtendrán las sentencias SQL de los insert, de los PrimaryItems, dimensiones, atributos de dimensión y relaciones entre dimensiones y atributos.

#### 5.4.2 Hito 2

El objetivo de este hito es la extracción de datos directamente de la instancia de la taxonomía para conseguir los **contextos** y **hechos** de dicha taxonomía.

Como resultado del hito anterior, este hito se va a desarrollar tanto para FINREP 2012 como para Solvency II.

El programa que se obtendrá de este hito será `xbrl002.vbs`.

##### 5.4.2.1 FINREP 2012

Antes de empezar a programar vamos a estudiar cómo están estructurados los datos en el fichero instancia de FINREP 2012. En el fichero instancia se diferencian tres partes que son: la cabecera, los contextos y los hechos.

La cabecera o Header es la siguiente, donde se puede ver información del Eschema XML de la instancia y su referencia.

```
<xbrli:xbrl xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="?
xmlns:dCU="http://www.eurofiling.info/dCU" xmlns:dCT="http://www.eurofiling.info/dCT" >
xmlns:dEC="http://www.eurofiling.info/dEC" xmlns:dBA="http://www.eurofiling.info/dBA" >
xmlns:nonnum="http://www.xbrl.org/dtr/type/non-numeric" xmlns:dRT="http://www.eurofiling.info/dRT" >
xmlns:dCL="http://www.eurofiling.info/dCL" xmlns:dRP="http://www.eurofiling.info/dRP" >
xmlns:dAT="http://www.eurofiling.info/dAT" xmlns:base="http://www.eurofiling.info/base"
xmlns:dsict="http://www.eurofiling.info/dsict" xmlns:dim="http://www.eurofiling.info/dim" >
xmlns:num="http://www.xbrl.org/dtr/type/numeric" xmlns:xlink="http://www.w3.org/1999/xlink" >
xmlns:iso4217="http://www.xbrl.org/2003/iso4217" xmlns:dGA="http://www.eurofiling.info/dGA" >
xmlns:xbrldi="http://xbrl.org/2006/xbrldi" xmlns:dMA="http://www.eurofiling.info/dMA" >
<link:schemaRef xlink:type="simple" xlink:href="entry.xsd"/>
```

Ilustración 17: Cabecera de Instancia de FINREP 2012

A continuación se muestra una ilustración de cómo es el formato de un contexto, en el que tenemos el *id*, que será el nombre o identificador del contexto. Como vemos un contexto se divide en sí en tres partes: *entity*, *period* y *scenario*.

- En *entity* podemos encontrar el identificar del esquema utilizado.
- En *period* encontramos el periodo en el que ese contexto aplica, que puede ser un instante de tiempo o el periodo comprendido entre dos fechas.
- En *Scenario* están los atributos con sus dimensiones que el contexto tiene, que están esquematizados como *dimensión* y después para el nombre de la dimensión *dim* y el atributo de dimensión esta puesto como el *value* de la expresión XML.

```
</xbrli:context>
<xbrli:context id="e_x7_x20_x2_gt0d_le90d_x11_x1">
  <xbrli:entity>
    <xbrli:identifier scheme="http://scheme">abc</xbrli:identifier>
  </xbrli:entity>
  <xbrli:period>
    <xbrli:instant>2011-06-12</xbrli:instant>
  </xbrli:period>
  <xbrli:scenario>
    <xbrldi:explicitMember dimension="dim:AS">dCT:x7</xbrldi:explicitMember>
    <xbrldi:explicitMember dimension="dim:AT">dAT:x20</xbrldi:explicitMember>
    <xbrldi:explicitMember dimension="dim:CS">dSE:x2</xbrldi:explicitMember>
    <xbrldi:explicitMember dimension="dim:DL">dTI:gt0d_le90d</xbrldi:explicitMember>
    <xbrldi:explicitMember dimension="dim:PL">dPL:x11</xbrldi:explicitMember>
    <xbrldi:explicitMember dimension="dim:RS">dRS:x1</xbrldi:explicitMember>
  </xbrli:scenario>
</xbrli:context>
```

Ilustración 18: Contexto de instancia de FINREP 2012

Ahora se va a mostrar cómo está estructurado un hecho en la instancia de esta taxonomía, en el que cada línea corresponde a un hecho, todos estos están etiquetados con el nombre del concepto básico y tiene la información de sus decimales, la referencia al contexto al que hace referencia el hecho, la unidad monetaria que aplica y el *value* será el propio valor del hecho.

```
<base:mi1 decimals="0" contextRef="e_x22_x3_x4_x9_x14_x1" unitRef="EUR">4</base:mi1>
<base:mi1 decimals="0" contextRef="e_x22_x3_x9_x14_x1" unitRef="EUR">44</base:mi1>
<base:mi1 decimals="0" contextRef="e_x7_x3_emu_x14_eur_x14_x1" unitRef="EUR">15</base:mi1>
<base:mi1 decimals="0" contextRef="e_x7_x3_x2_x14_eur_x14_x1" unitRef="EUR">15</base:mi1>
<base:mi1 decimals="0" contextRef="e_x7_x3_emu_x14_x2_x14_x1" unitRef="EUR">15</base:mi1>
<base:mi1 decimals="0" contextRef="e_x7_x3_x2_x14_x2_x14_x1" unitRef="EUR">15</base:mi1>
<base:mi1 decimals="0" contextRef="e_x7_x3_eu_x14_x14_x1" unitRef="EUR">60</base:mi1>
<base:mi1 decimals="0" contextRef="e_x7_x3_x4_x14_x14_x1" unitRef="EUR">40</base:mi1>
```

Ilustración 19: Hecho de instancia de FINREP 2012

#### 5.4.2.1.1 Desarrollando el programa

En este apartado se va a describir como actual el programa xbrl002.vbs para la creación de las inserciones en FINREP 2012 directamente de la instancia.

1. Entrada de cuatro argumentos por parámetro (fichero instancia, fichero log, fichero SQL, y nombre base de datos)
2. Leemos cada línea del fichero de la aserción y así vamos comprobando en que estructura nos encontramos:

**Ponemos** header=1

**SI** header=1 y no estamos en <xbrli:context

**Insertar** en Header  
**SI** estamos en <xbrli:context>  
    **Ponemos** header=0  
    **Ponemos** context=1  
    **SI** tenemos "id="                      **Guardamos** ID del contexto  
    **SI** tenemos "entity"                  **Guardamos** ID de la entidad  
    **SI** tenemos "period"                  **Guardamos** period  
    **Insertar** contexto( id, idEntity, periodo)  
**SI** estamos en <xbrli:scenario>  
    **Buscar** dimensión=dim → **Guardar** dimensión  
    **Insertar** Relacion contexto, dimensión y value(atributo de dimensión)  
**SI** estamos en <xbrli:unit>  
    **SI** tenemos <xbrli:measure>  
        **Insertar** Unidad  
**SI** tenemos <base>  
    **Insertar** Hecho con concepto básico, decimal, contexto, unidad, value

#### 5.4.2.2 Solvency II

En este apartado del hito 2 se estudiará cómo es el fichero instancia para la taxonomía de Solvency II y si con el programa creado anteriormente para FINREP 2012 obtenemos la cabecera, contextos y hechos de Solvency II.

En el primer paso ejecutamos el programa xbrl002.vbs con el fichero instancia de Solvency II y se consiguen los siguientes resultados:

- Se obtienen las inserciones de la cabecera correctamente.
- Se obtienen las inserciones de los contextos, pero podemos ver que estos contextos son obtenidos con errores.
- No se obtiene ninguna inserción para los hechos de la taxonomía.

Visto que los resultados no son del todo correctos pero están cerca de la solución, se estudiará el fichero instancia XBRL de Solvency II.

La cabecera del fichero instancia es la siguiente, donde se ve la referencia al esquema

```
<xbrli:xbrl xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xbrl.org/2006/xbrldt
xmlns:entry="http://entry" xmlns:gen="http://xbrl.org/2008/generic" xmlns:xbrldt="http://xbrl.org/2005/xbrldt" x
xmlns:xl="http://www.xbrl.org/2003/XLink" xmlns:d-soc="http://www.eiopa.europa.eu/fr/supervi/reporting/solvency2
xmlns:formula-roles="http://xbrl.org/formula/conformance/roles"
xmlns:t-BS_C1="http://www.eiopa.europa.eu/fr/supervi/reporting/solvency2/t-BS_C1_BalanceSheetTemplate"
xmlns:d-per="http://www.eiopa.europa.eu/fr/supervi/reporting/solvency2/d-per_Periodicity" xmlns:link="http://www
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:p-BS_C1="http://www.eiopa.europa.eu/fr/supervi/reporting/solven
xmlns:iso4217="http://www.xbrl.org/2003/iso4217" xmlns:xbrldi="http://xbrl.org/2006/xbrldi" xmlns:label="http://
xmlns:p-sty="http://www.eiopa.europa.eu/fr/supervi/reporting/solvency2/primary-types" xmlns:xbrli="http://www.xb
<link:schemaRef xlink:type="simple" xlink:href="sol2/t/entry.xsd"/>
```

### Ilustración 20: Cabecera de instancia Solvency II

A continuación se muestra una ilustración de cómo es el formato de un contexto en el que se encuentra el identificador del contexto y se diferencian también tres partes, *entity*, *period* y *scenario*.

- En *entity* podemos encontrar el identificador del esquema utilizado.
- En *period* encontramos el periodo en el que ese contexto aplica, que puede ser un instante de tiempo o el periodo comprendido entre dos fechas.
- En *Scenario* están los atributos con sus dimensiones que el contexto tiene. La dimensión de identifica por *dimensión*= y el atributo de dimensión será el *value*, como se ve muy parecido a FINREP 2012.

```
<xbrli:context id="Context_Instant_Quarterly_Solo">
  <xbrli:entity>
    <xbrli:identifier scheme="http://draft.lei.org/">123456</xbrli:identifier>
  </xbrli:entity>
  <xbrli:period>
    <xbrli:instant>2012-06-30</xbrli:instant>
  </xbrli:period>
  <xbrli:scenario>
    <xbrldi:explicitMember dimension="d-per:PeriodicityDimension">d-per:Quarterly</xbrldi:explicitMember>
    <xbrldi:explicitMember dimension="d-soc:SoloOrGroupDimension">d-soc:Solo</xbrldi:explicitMember>
  </xbrli:scenario>
</xbrli:context>
```

### Ilustración 21: Contexto de instancia de Solvency II

Ahora se va a mostrar cómo está estructurado un hecho en la instancia de esta taxonomía en la que cada línea corresponde a un hecho de la taxonomía, todos estos empiezan por su concepto básico identificados por *p-*, con sus decimales, la referencia al contexto, la referencia la unidad y el valor

```
<p-BS_C1:AS17A decimals="0" contextRef="Context_Instant_Quarterly_Solo" unitRef="EURO">42000</p-BS_C1:AS17A>
<p-BS_C1:AS18 decimals="0" contextRef="Context_Instant_Quarterly_Solo" unitRef="EURO">29655</p-BS_C1:AS18>
<p-BS_C1:AS17 decimals="0" contextRef="Context_Instant_Quarterly_Solo" unitRef="EURO">12345</p-BS_C1:AS17>
<p-BS_C1:AS18 decimals="0" contextRef="Context_Instant_Yearly_Solo" unitRef="EURO">69000</p-BS_C1:AS18>
<p-BS_C1:AS17 decimals="0" contextRef="Context_Instant_Yearly_Solo" unitRef="EURO">666</p-BS_C1:AS17>
<p-BS_C1:AS17A decimals="0" contextRef="Context_Instant_Yearly_Solo" unitRef="EURO">100000</p-BS_C1:AS17A>
```

### Ilustración 22: Hechos de Instancia de Solvency II

#### 5.4.2.2.1 Modificando el programa xbrl002.vbs

Analizado el fichero instancia, se procede a modificar el programa VBS generado anteriormente para FINREP 2012 de manera que también sea útil para Solvency II.

1. Entrada de cuatro argumentos por parámetro (fichero instancia, fichero log, fichero SQL, y nombre base de datos)

2. Leemos cada línea del fichero de la aserción y así vamos comprobando en que estructura nos encontramos:

**Ponemos** header=1

**SI** header=1 y no estamos en <xbrli:context

**Insertar** en Header

**SI** estamos en <xbrli:context>

**Ponemos** header=0

**Ponemos** context=1

**SI** tenemos "id="

**Guardamos** ID del contexto

**SI** tenemos "entity"

**Guardamos** ID de la entidad

**SI** tenemos "period"

**Guardamos** period

**Insertar** contexto( id, idEntity, periodo)

**SI** estamos en <xbrli:scenario>

**Buscar** dimensión → **Guardar** dimensión

**Insertar** Relacion contexto, dimensión y value(atributo de dimensión)

**SI** estamos en <xbrli:unit

**SI** tenemos <xbrli:measure

**Insertar** Unidad

**SI** tenemos <p-

**Insertar** Hecho con concepto básico, decimal, contexto, unidad, value

#### 5.4.2.4 Manual para el usuario

Para la ejecución de este programa se utilizará el archivo xbrl002.cmd, en el cual se debe poner la ruta de los ficheros y los parámetros del programa sabiendo que: el primero corresponde al fichero instancia de la taxonomía, el segundo al fichero log, el tercero al fichero de inserciones y el cuarto al nombre de la base de datos.

El resultado será un archivo SQL con las inserciones obtenidas. Para la ejecución de este fichero SQL en la base de datos, esta base de datos deberá tener las tablas ya creadas y ejecutado el fichero de inserciones del resultado de la ejecución del programa xbrl001.vbs del hito 1 para la misma taxonomía. De la ejecución del fichero SQL solución de xbrl002.vbs se obtendrán las inserciones de la cabecera, contextos, hechos y relaciones correspondientes. Y ya estará todos los datos de la taxonomía pasados a SQL.

#### 5.4.3 Hito 3

Como se ha explicado brevemente en el estudio y alcance del hito 3, en esta parte ya tenemos toda la base de datos creada y todas las inserciones a la base de datos hechas, por ello en esta parte trataremos las fórmulas de la taxonomía. Las formulas de la taxonomía son unos elementos matemáticos que deben ser cumplidos para la validación de la taxonomía, es decir, para saber si los datos alojados en la taxonomía son correctos.

Las formulas de las taxonomías pueden ser muy diferentes unas de otras, cada una trata una serie de datos y una serie de conjuntos. Estas fórmulas no solo validan los valores y los tipos de los datos sino que también sirven para documentar las relaciones de cálculo complejas ente conceptos.

Para este tercer y último hito se tratará primero la taxonomía de Solvency II, dado que esta taxonomía se considera más sencilla que FINREP 2012 y para empezar con validaciones será más sencillo en este orden.

El programa que obtendremos de este desarrollo será xbrl003.vbs

#### 5.4.3.1 Solvency II

El tratamiento de fórmulas en esta taxonomía será lo más genérico posible, para Solvency II sólo tenemos una fórmula de validación, sobre todo dado a que esta taxonomía es una prueba o versión inicial de la taxonomía de Solvency II que todavía no está publicada.

Para empezar a desarrollar nuestro programa en VBS que genere un programa en SQL de validación de la base de datos, se debe estudiar cómo es una formula en XBRL tratada primero con Arelle para simplificar el problema.

La fórmula que nos encontramos en Solvency II es una aserción llamada AS17A

##### 5.4.3.1.1 Assertion AS17A

Esta aserción se nos presenta en la siguiente ilustración.

```
Formula object,,,Label,Cover,Com-ple-ment,Bind as se-quence,Expression
valueAssertion,,,Assertion_AS17A,,,abs( ($va_0) - (($va_1 + $va_2)) ) le 3000
,factVariable $va_0,,va_0,,,false,fallbackValue =0
,,conceptName,Concept_BS_C1_AS17A,true,false,,p-BS_C1:AS17A
,,generalMeasures,unit_filter,true,false,,true()
,,period,period_filter,true,false,,true()
,factVariable $va_1,,va_1,,,false,fallbackValue =0
,,generalMeasures,unit_filter,true,false,,true()
,,conceptName,Concept_BS_C1_AS17,true,false,,p-BS_C1:AS17
,,period,period_filter,true,false,,true()
,factVariable $va_2,,va_2,,,false,fallbackValue =0
,,conceptName,Concept_BS_C1_AS18,true,false,,p-BS_C1:AS18
,,generalMeasures,unit_filter,true,false,,true()
,,period,period_filter,true,false,,true()
,precondition,,Assertion_AS17A_existence_precondition,,,,false() or $va_1 instance of node()* or $va_2 instance of node()*
```

Ilustración 23: Solvency II Aserción AS17A

La estructura de las formulas de validación de XBRL siguen el mismo formato:

- Lo primero es el tipo de formula, en nuestro caso “Formula object”, “expression”
- A continuación la asercion son su nombre y la fórmula matemática que se debe cumplir.
- Despues para cada variable tenemos:
  1. fallBackValue valor por defecto si la variable es null.
  2. El nombre del concepto básico que aplica a esa variable.
  3. Su unidad
  4. Su periodo
- Y por último la precondition, esa condición que se debe cumplir para poder empezar a validar la taxonomía.



Esta aserción ningún valor de contexto, dimensión o atributo de dimensión en el que se deba cumplir la validación con lo cual esta fórmula debe ser cumplida por todos los contextos de la taxonomía. Estos contextos, por supuesto, deberán cumplir la precondition para poder ser validados.

#### 5.5.3.1.1.1 Creación del programa *xbrl003.vbs*

Como se ha comentado antes, partimos de un programa creado para la taxonomía FINREP 2008 informe 6610, y se usaran tipos de formulas para su tratamiento a SQL. Usando este programa trataremos de validar la formula.

En el programa dado existen los siguientes tipos:

1. Type I: Trata fórmulas con un concepto, sin dimensiones ni parámetros.
2. Type II: Trata fórmula con más de un conceptos y algún parámetro.
3. Type III: Trata fórmulas con una dimensión, una aserción y sin conceptos ni parámetros.
4. Type IV: Trata fórmulas con una o más dimensiones, sin conceptos ni parámetros y con más de una variable.
5. Type V: Trata formulas con más de un concepto, con más de un filtro por contexto y que tenga una sentencia de aserción a true.

La aserción AS17A debe evaluarse para todos los contexto, y esta fórmula tiene tres conceptos a tratar, con lo cual no hay ningún tipo que esta fórmula pueda aplicar, se creará el tipo nuevo, type 6, que tratará este tipo de fórmulas y se modificará el programa para que pueda ser compatible con Solvency II.

Ahora vamos a describir cómo actúa el programa *xbrl003.vbs* para trata esta fórmula de Solvency II.

1. Entrada de cinco argumentos por parámetro (fichero aserción, fichero log, fichero SQL, fichero validación y nombre base de datos)
2. Leemos cada línea del fichero de la aserción y así vamos procesando de manera individualizada los datos que nos interesan de cada línea.
  - SI** contiene "formula" estamos en la primera línea → `newAssertionGroup=1`
  - SI** contiene "valueAssertion" estamos en la línea de la aserción →
    - Inicializamos los valores que trataremos para la aserción
    - Obtenemos el nombre de la aserción
    - Obtenemos la formula de la aserción, analizando y remplazando valores ( \$.@ )
  - SI** contiene "factVariable" estamos en la línea de valores por defecto
    - Guardamos el valor de `fallbackValue` con el nombre de la variable asignado
  - SI** contiene "conceptName" estamos en la línea del nombre del concepto
    - Guardamos el nombre del concepto con el nombre de la variable asignado
  - SI** ha pasado por los anteriores, estamos en la precondition
    - Obtenemos la precondition remplazando valores ( \$.@ )
    - Evaluamos la precondition para saber variables afectadas y tratamiento
3. Después de leer cada línea y tratar sus datos, vamos a asignar el tipo de formula o aserción que tenemos.



- SI** no tenemos dimensión ni filtros ni contextos y variables >1 → tipo=6
4. Se realiza la rutina de tipo 6.
- Declare** nombre, variable y fallBackValue por cada variable
- Cursor** realiza un cursor que recorra todos los contextos
- Select** traemos los valores correspondientes a cada concepto básico por cada contexto
- Precondicion** tratamos la preciondición, si se cumple:
- SI** el valor de la variable es null → asignamos fallBackValue
- Realizamos la fórmula de validación**
- SI** correcta → "Validación correcta"
- SI** incorrecta → "Validación incorrecta"
- SI** no cumple precondición → "Precondición Fallida"

#### 5.4.3.2 FINREP 2012

En tratamiento de fórmulas o aserciones en esta taxonomía será lo más genérico posible.

Como se ha comentado anteriormente en el estudio del Hito3, se tratan primero las aserciones con Arelle para simplificar la complejidad y así tratar los datos en texto plano. Haciendo este proceso para FINREP 2012 obtenemos doce aserciones que son las siguientes:

- t03\_hr01
- t03\_hr02
- t03\_t05\_ct01
- t03\_vr01
- t03\_vr02
- t03\_vr03
- t05\_hr01
- t05\_hr02
- t05\_vr01
- t05\_vr02
- t05\_vr03
- t07\_vr01

Comenzaremos tratando la primera de ellas y a continuación iremos estudiando una a una las diferentes aserciones.

##### 5.4.3.2.1 Assertion t03\_hr01

Esta aserción se nos presenta en las ilustraciones siguiente.

```
valueAssertion,,,,,t03_hr01,,, $CarryingAmount = $UnimpairedAndNotPastDue +  
$UnimpairedAndPastDueUnder90Days + $UnimpairedAndPastDueBetween90and180Days +  
$UnimpairedAndPastDueBetween180and1Year + $UnimpairedAndPastDueOver1Year +  
$ImpairedNet  
,andFilter,,,,andFilter_8,,,  
,,conceptName,,,,conceptName,,,,base:mi1  
,,explicitDimension,,,explicitDimension_15,,,dim:PL  
,,explicitDimension,,,explicitDimension_16,,,dim:RS  
,,orFilter,,,,orFilter_3,,,  
,,andFilter,,,andFilter_5,,,  
,,,,explicitDimension,,,explicitDimension_17,,,dim:AS  
,,,,explicitDimension,,,explicitDimension_22,,,dim:CS  
member: dSE:x14  
linkrole: http://www.eurofiling.info/dsi/ct_04"  
,,andFilter,,,andFilter_6,,,  
,,,,explicitDimension,,,explicitDimension_18,,,dim:AS  
,,,,explicitDimension,,,explicitDimension_23,,,dim:CS  
member: dSE:x1  
linkrole: http://www.eurofiling.info/dsi/ct_02"  
,,andFilter,,,andFilter_7,,,  
,,,,explicitDimension,,,explicitDimension_21,,,dim:AS  
,,,,explicitDimension,,,explicitDimension_19,,,dim:CS
```

Ilustración 24: FINREP 2012 aserción t03\_hr01 fórmula y filtros

En esta primera ilustración del fichero instancia tratado por Arelle podemos observar los siguientes campos o valores:

- ValueAssertion son su nombre y la fórmula matemática que se debe cumplir.
- Después observamos que viene los filters, lo que quiere decir que esta validación depende de unos filtros. Primero se ve Filter8, con un **and** y después filtro 3 con un **or** en el que incluye los filtros 5, 6 y 7.

Este filtros que se han encontrado en la aserción requieren un tratamiento especial, podemos tratar estos filtros como grupos de variables que deben cumplir la validación. Los grupos están formados por:

- Grupo 1 → filtro 8 y 5
- Grupo 2 → filtro 8 y 6
- Grupo 3 → filtro 8 y 7.

```
,factVariable $UnimpairedAndNotPastDue,,,,factVariable_7,,,false,fallbackValue =0
,,explicitDimension,,,,explicitDimension_9,true,false,,dim:AT
,,explicitDimension,,,,explicitDimension_10,true,false,,dim:DL
,factVariable
$UnimpairedAndPastDueUnder90Days,,,,factVariable_8,,,false,fallbackValue =0
,,explicitDimension,,,,explicitDimension_9,true,false,,dim:AT
,,explicitDimension,,,,explicitDimension_11,true,false,,dim:DL
,factVariable
$UnimpairedAndPastDueBetween90and180Days,,,,factVariable_9,,,false,fallbackValue =0
,,explicitDimension,,,,explicitDimension_9,true,false,,dim:AT
,,explicitDimension,,,,explicitDimension_12,true,false,,dim:DL
,factVariable
$UnimpairedAndPastDueBetween180and1Year,,,,factVariable_10,,,false,fallbackValue =0
,,explicitDimension,,,,explicitDimension_9,true,false,,dim:AT
,,explicitDimension,,,,explicitDimension_13,true,false,,dim:DL
,factVariable
$UnimpairedAndPastDueOver1Year,,,,factVariable_11,,,false,fallbackValue =0
,,explicitDimension,,,,explicitDimension_9,true,false,,dim:AT
,,explicitDimension,,,,explicitDimension_14,true,false,,dim:DL
,factVariable $ImpairedNet,,,,factVariable_12,,,false,fallbackValue =0
,,explicitDimension,,,,explicitDimension_4,true,false,,dim:AT
,factVariable $CarryingAmount,,,,factVariable_13,,,false,fallbackValue =0
,,explicitDimension,,,,explicitDimension_3,true,false,,dim:AT
```

Ilustración 25: FINREP 2012 aserción t03\_hr01 variables

En la segunda ilustración de esta aserción observamos las variables con sus valores y sus FallbackValue. Cada una de estas variables sigue la siguiente estructura:

- Nombre de variable y valor de FallbackValue
- La dimensión o dimensiones de las que está compuesta esa variable.

Si se observa con detenimiento cada una de estas ilustraciones observamos que tenemos las dimensiones pero no tenemos los atributos de dimensión a los que hace referencia, con lo cual se procede a mirar en el fichero xml de la validación para comprobar donde está el error en los nombres de los atributos de dimensión.

```
<df:explicitDimension xlink:type="resource" xlink:label="explicitDimension_15" id=
"Filter_Portfolio_MeasuredAtAmortisedCost">
  <df:dimension>
    <df:qname>dim:PL</df:qname>
  </df:dimension>
  <df:member>
    <df:qname>dPL:x11</df:qname>
  </df:member>
</df:explicitDimension>
```

Ilustración 26: FINREP 2012 aserción t03\_hr01 xml

Cogiendo como ejemplo el fichero t03-form.xml, se puede observar que para una dimensión viene asociado un atributo de dimensión, en este caso explicitDimension\_15, con lo cual, con esta comprobación se determina que Arelle genera mal los fichero de validación para FINREP 2012 al no insertar los atributos de dimensión.

En esta parte se presentan diferentes alternativas para solucionar este problema del fichero de aserciones obtenido con Arelle:

La primera sería tratar con el fichero de aserciones de Arelle y cuando lleguen los atributos de dimensión ir a buscarlos al fichero XML de validación. Esta alternativa no sé implementará ya que sería necesario saber cómo se llama el fichero de validación donde se encuentra esta validación.

La segunda sería tratar directamente con el fichero XML de aserción para así tener todos los metadatos. Esta alternativa no se llevará a cabo ya que lo que se quiere y lo que se pretende en este proyecto es el tratamiento de cualquier taxonomía y como se ve anteriormente para Solvency II funciona correctamente.

La tercera alternativa consistiría en añadir los atributos de dimensión nosotros mismos en el fichero que se obtiene de Arelle y así poder tratar la fórmula de validación. Esta alternativa sería provisional ya que se espera que Arelle mejore la forma de tratar sus datos para adecuarlo a FINREP 2012.

Dado este problema de entrada en el fichero de aserciones obtenido por Arelle, solo vamos a tratar esta primera formula, t03\_hr01, dejando pendientes las otras fórmulas de FINREP 2012.

Esta aserción ningún valor de contexto, dimensión o atributo de dimensión en el que se deba cumplir la validación con lo cual esta fórmula debe ser cumplida por todos los contextos de la taxonomía. Estos contextos, por supuesto, deberán cumplir la precondition para poder ser validados.

#### 5.5.3.1.2 Modificación del fichero de aserción

Antes de ponerse con la modificación del programa VBS de validación, vamos a introducir los atributos de dimensión en el fichero de aserciones.

Se ha estudiado que esta fórmula está tratada por filtros con dimensiones explícitas, con lo cual vamos a investigar cómo trata Arelle una fórmula de validación con dimensiones explícitas para FINREP 2008 en su informe 6610.

```
,valueAssertion,,val_b-or-4,,, $activosFinancierosDisponiblesVentaPromemoriaPrestadosEnGarantia le
$activosFinancierosDisponiblesVta
,,explicitDimension,,fil_DistribucionDim_TotalSectoriald1898e344,,,es-be-d-FR-dist:DistribucionDimension
,,factVariable
$activosFinancierosDisponiblesVentaPromemoriaPrestadosEnGarantia,,var_es-be-p-FINREP_ActivosFinancierosDis
poniblesVentaPromemoriaPrestadosEnGarantiad1898e348,,,false,fallbackValue =0
,,,conceptName,fil_es-be-p-FINREP_ActivosFinancierosDisponiblesVentaPromemoriaPrestadosEnGarantiad1898e349
,true,false,,es-be-p-FINREP:ActivosFinancierosDisponiblesVentaPromemoriaPrestadosEnGarantia
,,factVariable
$activosFinancierosDisponiblesVta,,var_ifrs-gp_AvailableForSaleFinancialAssetsTotald1898e353,,,false,fallb
ackValue =0
,,,conceptName,fil_ifrs-gp_AvailableForSaleFinancialAssetsTotald1898e354,true,false,,ifrs-gp:AvailableForS
aleFinancialAssetsTotal
,valueAssertion,,val_b-or-5,,, $inversionesCrediticiasPromemoriaPrestadosEnGarantia le
$inversionesCrediticias
,,explicitDimension,,fil_DistribucionDim_TotalSectoriald1898e360,,,es-be-d-FR-dist:DistribucionDimension
,,factVariable
```

Ilustración 27: Aserción FINREP 2006 informe 6610

En la imagen expuesta se tienen dos dimensiones explícitas y se observa que la estructura que sigue es:

explicitDimension-----atributo de dimensión ----- dimensión

Mientras que en el fichero para FINREP 2012, que lo podemos ver en la ilustración 25, es:

explicitDimension-----número de explicitDimension ----- dimensión

De esta manera tenemos en cuenta que en FINREP 2012 se introduce el número de dimensión mientras que en la versión 2008 se pone el atributo de dimensión, con lo cual para poder desarrollar esta primera fórmula se cambiará el valor del número de dimensión por el nombre del atributo de dimensión. Estos valores de los atributos de dimensión los podemos encontrar en el fichero t03-form.xml de la taxonomía, y para encontrar los valores correctos de los atributos solo hace falta buscar por el número de dimensión explícita.

Siguiendo estos pasos el fichero de aserciones obtenido de Arelle queda de la siguiente forma:

```
valueAssertion,,,,,t03_hr01,,, $CarryingAmount = $UnimpairedAndNotPastDue +
$UnimpairedAndPastDueUnder90Days + $UnimpairedAndPastDueBetween90and180Days +
$UnimpairedAndPastDueBetween180and1Year + $UnimpairedAndPastDueOver1Year + $ImpairedNet
, andFilter,,,, andFilter_8,,,,
,, conceptName,,,, conceptName,,,, base:mi1
,, explicitDimension,,,, dPL:x11,,,, dim:PL
,, explicitDimension,,,, dRS:x1,,,, dim:RS
,, orFilter,,,, orFilter_3,,,,
,, andFilter,,,, andFilter_5,,,,
,, explicitDimension,,,, dCT:x7,,,, dim:AS
,, explicitDimension,,,, dSE:x14,,,, "dim:CS
member: dSE:x14
linkrole: http://www.eurofiling.info/dsi/ct_04"
,, andFilter,,,, andFilter_6,,,,
,, explicitDimension,,,, dCT:x22,,,, dim:AS
,, explicitDimension,,,, dSE:x1,,,, "dim:CS
member: dSE:x1
linkrole: http://www.eurofiling.info/dsi/ct_02"
,, andFilter,,,, andFilter_7,,,,
,, explicitDimension,,,, dCT:x3,,,, dim:AS
,, explicitDimension,,,, dSE:x1,,,, dim:CS
factVariable $UnimpairedAndNotPastDue,,,, factVariable_7,,, false, fallbackValue =0
,, explicitDimension,,,, dAT:x20, true, false, dim:AT
,, explicitDimension,,,, dTI:eq0d, true, false, dim:DL
factVariable $UnimpairedAndPastDueUnder90Days,,,, factVariable_8,,, false, fallbackValue =0
,, explicitDimension,,,, dAT:x20, true, false, dim:AT
,, explicitDimension,,,, dTI:gt0d_le90d, true, false, dim:DL
```

Ilustración 28: FINREP 2012 aserción t03\_hr01 modificado

Para no modificar el fichero de aserciones de Arelle, esta aserción en la que están los atributos de dimensión para 103\_hr01 se guardarán en un fichero diferente y será el que se utilizará para desarrollar el programa.

#### 5.5.3.1.1.3 Modificación del programa xbrl003.vbs

En el apartado de la creación del programa xbrl003.vbs se ha comentado como está estructurado este programa, esta estructura es por tipos de manera que cada aserción dependiendo de sus datos pertenece a un tipo o a otro, los tipos que tenemos son los siguientes ya añadiendo el nuevo tipo creado para la aserción de validación de Solvency II:

1. Type I: Trata fórmulas con un concepto, sin dimensiones ni parámetros.
2. Type II: Trata fórmula con más de un concepto y algún parámetro.

3. Type III: Trata fórmulas con una dimensión, una aserción y sin conceptos ni parámetros.
4. Type IV: Trata fórmulas con una o más dimensiones, sin conceptos ni parámetros y con más de una variable.
5. Type V: Trata fórmulas con más de un concepto, con más de un filtro por contexto y que tenga una sentencia de aserción a true.
6. Type VI: Trata fórmulas con más de un concepto y para todos los contextos de la taxonomía.

La aserción t03\_hr01 tiene una sentencia de aserción, un solo concepto, no tiene número de parámetros y tiene más de una dimensión explícita. Ninguno de estos tipos de aserciones de adecua a esta aserción con lo cual se va a crear un nuevo tipo llamado Type VII, que reúna las fórmulas que tengan estos datos.

Vamos a describir cómo actúa este tipo de aserciones para FINREP 2012:

1. Entrada de cinco argumentos por parámetro (fichero aserción, fichero log, fichero SQL, fichero validación y nombre base de datos)
2. Leemos cada línea del fichero de la aserción y así vamos procesando de manera individualizada los datos que nos interesan de cada línea.  
**SI** contiene "formula" estamos en la primera línea → newAssertionGroup=1  
**SI** contiene "valueAssertion" estamos en la línea de la aserción →  
    Inicializamos los valores que trataremos para la aserción  
    Obtenemos el nombre de la aserción  
    Obtenemos la formula de la aserción, analizando y remplazando valores ( \$..@)  
**SI** contiene "filter" estamos en los filtros de la aserción  
    Vamos a leer los filtros.  
    Guardamos el atributo de dimensión y la dimensión de este filtro.  
**SI** contiene "factVariable" estamos en la línea de valores por defecto  
    Guardamos el valor de fallbackValue con el nombre de la variable asignado  
**SI** contiene hemos estado en "factVariable" y tenemos un filtro  
    Guardamos los valores que correspondan de conceto, atributo y dimensión con el nombre de la variable
3. Después de leer cada línea y tratar sus datos, vamos a asignar el tipo de formula o aserción que tenemos.  
**SI** tenemos aserción, concepto=1, parentros =0 y filtrosdimension>1 → Type VII
4. Se realiza la rutina de tipo 7.  
**Declare** nombre, variable y fallBackValue por cada variable  
Para cada filtro o grupo general:  
    Por cada variable:  
        **Select** para obtener el hecho, que engloba al grupo y a la variable  
        **Guardar** valor de fallBackValue.  
        **SI** variable no tiene valor y fallBackValue no es null → valor= fallBackValue  
        **SINO** → "Error en variables"  
    **Realizamos la fórmula de validación**  
        **SI correcta** → "Validación Correcta"

**SINO → “Validación Incorrecta”**

#### **5.4.3.3 Manual de usuario**

Para la ejecución de este programa se utilizará el archivo xbrl003.cmd, en el cual se debe poner la ruta de los ficheros y los parámetros del programa sabiendo que: el primero corresponde al fichero de la fórmula, el segundo al fichero log, el tercero al fichero de validación, el cuarto al fichero de procedure y el quinto al nombre de la base de datos.

Para poder ejecutar este proyecto en SQL, deben estar creada la base de datos con sus tablas e insertados todos los datos obtenidos en los hitos anteriores, ya que estamos tratando un programa de validación de base de datos.

Cuando se ejecute el programa solución en la base de datos saldrán mensajes de los contextos analizados y su resultado, si están validados, si falló la fórmula de validación o si no cumplen la precondition en el caso de que exista.

## 6. PRUEBAS DE LA SOLUCIÓN TÉCNICA

### 6.1 Pruebas de la solución técnica

En este apartado se realizan y crean las pruebas realizadas sobre la solución técnica al problema, es decir, sobre los programas creados en VBScript. Las pruebas serán genéricas para cualquier taxonomía y base de datos.

Estas pruebas al igual que los apartados anteriores de este proyecto se separarán en hitos, los tres hitos del proyecto que además coinciden con los tres subsistemas de desarrollo.

El formato de las pruebas será el siguiente:

<b>ID</b>	PR-TEC-XX
<b>OBJETIVO</b>	Describe el objetivo de esta prueba.
<b>SUBSISTEMAS</b>	Subsistema al que pertenece esta prueba
<b>ENTRADA</b>	Datos a insertar en la prueba
<b>SALIDA</b>	Datos de salida de la prueba
<b>ESTADO</b>	Estado de la prueba, puede ser Aceptada/Rechazada

Tabla 92: Plantilla pruebas diseño técnico

El identificador, se nombrará de la siguiente forma **PR-TEC-XX**, donde PR significa prueba, TEC solución técnica y XX corresponde con la numeración secuencial de las pruebas de la solución técnica.

#### 6.1.1 Hito 1

<b>ID</b>	PR-TEC-01
<b>OBJETIVO</b>	Verificar que con un formato correcto y completo se obtiene todas las inserciones en el fichero de salida.
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>ENTRADA</b>	Fichero tomado de Arelle.
<b>SALIDA</b>	Fichero de inserciones con la el nombre de la base de datos a usar, los primary ítems, dimensiones, atributos de dimensión y relación entre dimensiones y atributos
<b>ESTADO</b>	Aceptada

Tabla 93: Prueba PR-TEC-01



<b>ID</b>	PR-TEC-02
<b>OBJETIVO</b>	Verificar que con un fichero vacío el programa no da error y no genera inserciones.
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>ENTRADA</b>	Fichero vacío
<b>SALIDA</b>	Fichero SQL solo con el nombre de la base de datos.
<b>ESTADO</b>	Aceptada

Tabla 94: Prueba PR-TEC-02

<b>ID</b>	PR-TEC-03
<b>OBJETIVO</b>	Verificar que con un fichero con solo conceptos básicos, inserta los conceptos básicos en el SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>ENTRADA</b>	Fichero sacado de Arelle en el que solo hay líneas para la inserción de conceptos básicos.
<b>SALIDA</b>	Fichero SQL con el nombre de base de datos a utilizar y las inserciones de los conceptos básicos.
<b>ESTADO</b>	Aceptada

Tabla 95: Prueba PR-TEC-03

<b>ID</b>	PR-TEC-04
<b>OBJETIVO</b>	Verificar que con un fichero con solo dimensiones y atributos de dimensión, se crean sus correspondientes en la base de datos.
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>ENTRADA</b>	Fichero sacado de Arelle con solo Dimensiones y Atributos de dimensión.
<b>SALIDA</b>	Fichero SQL en el que únicamente tenemos la base de datos a utilizar y las inserciones de las dimensiones, atributos y su relación
<b>ESTADO</b>	Aceptada

Tabla 96: Prueba PR-TEC-04

<b>ID</b>	PR-TEC-05
<b>OBJETIVO</b>	Verificar que si el fichero esta partido, no salta error en el programa.
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>ENTRADA</b>	Fichero sacado de Arelle que se corta a la mitad de los atributos de una dimensión
<b>SALIDA</b>	Fichero SQL con la creación de las inserciones hasta la final del fichero de entrada.
<b>ESTADO</b>	Aceptada

Tabla 97: Prueba PR-TEC-05

<b>ID</b>	PR-TEC-06
<b>OBJETIVO</b>	Verificar entrada de los parámetros de entrada H1
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls001_v3.vbs dimensiones_Solvency.txt xbrls001_v3_Solvency.log xbrls001_v3_insert_Solvency.sql TXBRL_FINREP2012</b>
<b>SALIDA</b>	Fichero de inserciones con la el nombre de la base de datos a usar, los primary ítems, dimensiones, atributos de dimensión y relación entre dimensiones y atributos
<b>ESTADO</b>	Aceptada

Tabla 98: Prueba PR-TEC-06

<b>ID</b>	PR-TEC-07
<b>OBJETIVO</b>	Verificar el parámetro de entrada “nombre base de datos” H1
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls001_v3.vbs dimensiones_Solvency.txt xbrls001_v3_Solvency.log xbrls001_v3_insert_Solvency.sql</b>
<b>SALIDA</b>	“Error en argumentos de entrada”  Fichero SQL sin nombre de base de datos y con todas las inserciones.

<b>ESTADO</b>	Aceptada
---------------	----------

Tabla 99: Prueba PR-TEC-07

<b>ID</b>	PR-TEC-08
<b>OBJETIVO</b>	Verificar el parámetro de entrada “Fichero dimensiones”
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls001_v3.vbs xbrls001_v3_Solvency.log xbrls001_v3_insert_Solvency.sql TXBRL</b>
<b>SALIDA</b>	“Error en argumentos de entrada”  Fichero erróneo.
<b>ESTADO</b>	Aceptada

Tabla 100: Prueba PR-TEC-08

<b>ID</b>	PR-TEC-09
<b>OBJETIVO</b>	Verificar el parámetro de entrada “Nombre fichero log” H1
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls001_v3.vbs dimensiones_Solvency.txt xbrls001_v3_insert_Solvency.sql TXBRL</b>
<b>SALIDA</b>	“Error en argumentos de entrada”  Fichero erróneo.
<b>ESTADO</b>	Aceptada

Tabla 101: Prueba PR-TEC-09

<b>ID</b>	PR-TEC-10
<b>OBJETIVO</b>	Verificar el parámetro de entrada “Nombre fichero inserciones” H1
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls001_v3.vbs dimensiones_Solvency.txt xbrls001_v3_Solvency.log TXBRL</b>
<b>SALIDA</b>	“Error en argumentos de entrada”  Fichero erróneo.
<b>ESTADO</b>	Aceptada

Tabla 102: Prueba PR-TEC-10

### 6.1.2 Hito 2

<b>ID</b>	PR-TEC-11
<b>OBJETIVO</b>	Verificar que con un formato correcto del fichero de instancia, instance.xbri, se obtiene todas las inserciones de salida.
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>ENTRADA</b>	Fichero instancia de la taxonomía.
<b>SALIDA</b>	Fichero de inserciones con la el nombre de la base de datos a usar, los contextos, los hechos, y la relaciones entre los contextos y los atributos de dimensión y dimensiones.
<b>ESTADO</b>	Aceptada

Tabla 103: Prueba PR-TEC-12

<b>ID</b>	PR-TEC-12
<b>OBJETIVO</b>	Verificar que con un fichero vacio de xbrl, el programa no da error y no genere inserciones.
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>ENTRADA</b>	Fichero xbrl vacio
<b>SALIDA</b>	Fichero SQL solo con la base de datos a uar.
<b>ESTADO</b>	Aceptada

Tabla 104: Prueba PR-TEC-12

<b>ID</b>	PR-TEC-13
<b>OBJETIVO</b>	Verificar que con un fichero solo con cabecera y contexto, se realizan las inserciones de estos.
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>ENTRADA</b>	Fichero de instancia xbrl solo con cabecera y contextos.
<b>SALIDA</b>	Fichero de inserciones SQL con las inserciones de la cabecera y los contextos
<b>ESTADO</b>	Aceptada

Tabla 105: Prueba PR-TEC-13

<b>ID</b>	PR-TEC-14
<b>OBJETIVO</b>	Verificar que con un fichero sin cabecera, no se realizan inserciones.
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>ENTRADA</b>	Fichero instancia sin cabecera.
<b>SALIDA</b>	Fichero SQL de inserciones erróneas.
<b>ESTADO</b>	Aceptada. Sin cabecera las inserciones no valen, no sabemos de que esquema proceden.

Tabla 106: Prueba PR-TEC-14

<b>ID</b>	PR-TEC-15
<b>OBJETIVO</b>	Verificar que si un fichero esta partido a mitad de los contextos, se realizan inserciones hasta donde existen enteros.
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>ENTRADA</b>	Fichero instancia que se corta a mitad de los contextos.
<b>SALIDA</b>	Error en el programa  Fichero SQL con las inserciones de la cabecera y los contextos escritos en la instancia.
<b>ESTADO</b>	Aceptada

Tabla 107: Prueba PR-TEC-15

<b>ID</b>	PR-TEC-16
<b>OBJETIVO</b>	Verificar entrada de los parámetros de entrada H2
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls002_v2.vbs instanceSolvency.xbrl xbrls002_Solvency.log xbrls002_insert_Solvency.sql TXBRL_SOLVENCY2</b>
<b>SALIDA</b>	Fichero de inserciones con la el nombre de la base de datos a usar, los contextos, los hechos, y la relaciones entre los contextos y los atributos de dimensión y dimensiones.
<b>ESTADO</b>	Aceptada

Tabla 108: Prueba PR-TEC-16

<b>ID</b>	PR-TEC-17
<b>OBJETIVO</b>	Verificar el parámetro de entrada “nombre base de datos” H2
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls002_v2.vbs instanceSolvency.xbrl xbrls002_Solvency.log xbrls002_insert_Solvency.sql</b>
<b>SALIDA</b>	“Error en argumentos de entrada”  Fichero SQL sin nombre de base de datos y con todas las inserciones.
<b>ESTADO</b>	Aceptada

Tabla 109: Prueba PR-TEC-17

<b>ID</b>	PR-TEC-18
<b>OBJETIVO</b>	Verificar el parámetro de entrada “Fichero instancia”
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls002_v2.vbs xbrls002_Solvency.log xbrls002_insert_Solvency.sql TXBRL_SOLVENCY2</b>
<b>SALIDA</b>	“Error en argumentos de entrada”  Fichero erróneo.

<b>ESTADO</b>	Aceptada
---------------	----------

Tabla 110: Prueba PR-TEC-18

<b>ID</b>	PR-TEC-19
<b>OBJETIVO</b>	Verificar el parámetro de entrada “Nombre fichero log” H2
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls002_v2.vbs instanceSolvency.xbrl xbrls002_insert_Solvency.sql TXBRL_SOLVENCY2</b>
<b>SALIDA</b>	“Error en argumentos de entrada”  Fichero erróneo.
<b>ESTADO</b>	Aceptada

Tabla 111: Prueba PR-TEC-19

<b>ID</b>	PR-TEC-20
<b>OBJETIVO</b>	Verificar el parámetro de entrada “Nombre fichero inserciones” H2
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls002_v2.vbs instanceSolvency.xbrl xbrls002_Solvency.log TXBRL_SOLVENCY2</b>
<b>SALIDA</b>	“Error en argumentos de entrada”  Fichero erróneo.
<b>ESTADO</b>	Aceptada

Tabla 112: Prueba PR-TEC-20

### 6.1.3 Hito 3

<b>ID</b>	PR-TEC-21
<b>OBJETIVO</b>	Verificar que con un formato correcto del fichero de aserciones, se genera el programa SQL de validación de la taxonomía.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>ENTRADA</b>	Fichero de la aserción de la taxonomía pasado por Arelle.
<b>SALIDA</b>	Fichero SQL con la el nombre de la base de datos a usar y los select de

	la validación de la taxonomía.
<b>ESTADO</b>	Aceptada

Tabla 113: Prueba PR-TEC-21

<b>ID</b>	PR-TEC-22
<b>OBJETIVO</b>	Verificar que con un fichero vacío, el programa no da error y no genere instrucciones SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>ENTRADA</b>	Fichero en formato plano vacío.
<b>SALIDA</b>	Fichero SQL con la el nombre de la base de datos a usar.
<b>ESTADO</b>	Aceptada

Tabla 114: Prueba PR-TEC-22

<b>ID</b>	PR-TEC-23
<b>OBJETIVO</b>	Verificar que con un fichero con solo la parte de valueAssertion no se crean rutinas SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>ENTRADA</b>	Fichero de aserciones solo en la línea de valueAssertion.
<b>SALIDA</b>	Fichero SQL con la el nombre de la base de datos a usar.
<b>ESTADO</b>	Aceptada

Tabla 115: Prueba PR-TEC-23

<b>ID</b>	PR-TEC-24
<b>OBJETIVO</b>	Verificar que con un fichero cortado a la mitad de la aserción, no da error el programa.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>ENTRADA</b>	Fichero de aserciones cortado en mitad de una aserción.
<b>SALIDA</b>	Fichero SQL con la el nombre de la base de datos a usar y un código SQL sin terminar.
<b>ESTADO</b>	Aceptada

Tabla 116: Prueba PR-TEC-24



<b>ID</b>	PR-TEC-25
<b>OBJETIVO</b>	Verificar parámetros de entrada H3
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls003_v3.vbs XBRLFormulaFINREP2012_v2.txt xbrls003_Finrep.log xbrls003_Finrep_procedure.sql xbrls003_Finrep_validation.sql TXBRL_FINREP2012</b>
<b>SALIDA</b>	Fichero SQL con la el nombre de la base de datos a usar y los select de la validación de la taxonomía.
<b>ESTADO</b>	Aceptada

Tabla 117: Prueba PR-TEC-25

<b>ID</b>	PR-TEC-26
<b>OBJETIVO</b>	Verificar el parámetro de entrada “nombre base de datos” H3
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls003_v3.vbs XBRLFormulaFINREP2012_v2.txt xbrls003_Finrep.log xbrls003_Finrep_procedure.sql xbrls003_Finrep_validation.sql</b>
<b>SALIDA</b>	“Error en argumentos de entrada”  Fichero SQL sin nombre de base de datos y los select de la validación.
<b>ESTADO</b>	Aceptada

Tabla 118: Prueba PR-TEC-26

<b>ID</b>	PR-TEC-27
<b>OBJETIVO</b>	Verificar el parámetro de entrada “Fichero aserciones”
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls003_v3.vbs xbrls003_Finrep.log xbrls003_Finrep_procedure.sql xbrls003_Finrep_validation.sql TXBRL_FINREP2012</b>
<b>SALIDA</b>	“Error en argumentos de entrada”

	Fichero Erróneo
<b>ESTADO</b>	Aceptada

Tabla 119: Prueba PR-TEC-27

<b>ID</b>	PR-TEC-28
<b>OBJETIVO</b>	Verificar el parámetro de entrada “nombre fichero procedure”
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls003_v3.vbs XBRLFormulaFINREP2012_v2.txt xbrls003_Finrep.log xbrls003_Finrep_validation.sql TXBRL_FINREP2012</b>
<b>SALIDA</b>	“Error en argumentos de entrada”  Fichero Erróneo
<b>ESTADO</b>	Aceptada

Tabla 120: Prueba PR-TEC-28

<b>ID</b>	PR-TEC-29
<b>OBJETIVO</b>	Verificar el parámetro de entrada “Nombre fichero log”
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls003_v3.vbs XBRLFormulaFINREP2012_v2.txt xbrls003_Finrep_procedure.sql xbrls003_Finrep_validation.sql TXBRL_FINREP2012</b>
<b>SALIDA</b>	“Error en argumentos de entrada”  Fichero Erróneo.
<b>ESTADO</b>	Aceptada

Tabla 121: Prueba PR-TEC-29

<b>ID</b>	PR-TEC-30
<b>OBJETIVO</b>	Verificar el parámetro de entrada “Nombre fichero validation”
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>ENTRADA</b>	Ejecución del programa según siguiente línea:  <b>cscript xbrls003_v3.vbs XBRLFormulaFINREP2012_v2.txt xbrls003_Finrep_procedure.sql TXBRL_FINREP2012</b>
<b>SALIDA</b>	“Error en argumentos de entrada”  Fichero Erróneo.
<b>ESTADO</b>	Aceptada

Tabla 122: Prueba PR-TEC-30

## 6.2 Pruebas de la solución obtenida

En este apartado se realizan y crean las pruebas realizadas sobre el fichero solución SQL, obtenidos de ejecutar el programa creado para la solución del problema, es decir, sobre los programas creados en VBScript.

Las pruebas se realizan sobre los ficheros solución, que son fichero SQL de inserciones para los Hito 1 y 2 y de comprobación y validación de la base de datos para el hito 3. Para el desarrollo de estas pruebas debe estar creada la base de datos, con sus tablas y relaciones, según el apartado “Elaboración de Especificaciones del Modelo Físico de Datos” y cumplidas las precondiciones de cada prueba.

Estas pruebas al igual que los apartados anteriores de este proyecto se separarán en hitos, los tres hitos del proyecto que además coinciden con los tres subsistemas de desarrollo.

El formato de las pruebas será el siguiente:

<b>ID</b>	PR-SOL-XX
<b>OBJETIVO</b>	Describe el objetivo de esta prueba.
<b>SUBSISTEMAS</b>	Subsistema al que pertenece esta prueba
<b>PRECONDICIONES</b>	Condiciones que tiene que haber para poder realizar la prueba
<b>ENTRADA</b>	Datos a insertar en la prueba
<b>SALIDA</b>	Datos de salida de la prueba
<b>ESTADO</b>	Estado de la prueba, puede ser Aceptada/Rechazada

Tabla 123: Plantilla Prueba solución

El identificador, se nombrará de la siguiente forma **PR-TEC-XX**, donde PR significa prueba, SOL solución obtenida y XX corresponde con la numeración secuencial de las pruebas de la solución obtenida.

### 6.2.1 Hito 1

<b>ID</b>	PR-SOL-01
<b>OBJETIVO</b>	Validar que la salida obtenida en la prueba PR-TEC-01, es correcta según lenguaje SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Nombre base de datos igual que el introducido en Salida prueba: <b>PR-TEC-01</b></li> </ol>
<b>ENTRADA</b>	Salida prueba: <b>PR-TEC-01</b>
<b>SALIDA</b>	<p>Command(s) completed successfully.</p> <ol style="list-style-type: none"> <li>1. Insertados los primaryItems</li> <li>2. Insertados dimensions</li> <li>3. Insertados DimAttributes</li> <li>4. Insertados Relation_Dim_AttrDim</li> </ol>
<b>ESTADO</b>	Aceptada

Tabla 124: Prueba PR-SOL-01

<b>ID</b>	PR-SOL-02
<b>OBJETIVO</b>	Validar que la salida obtenida en la prueba PR-TEC-02, es correcta según lenguaje SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Nombre base de datos igual que el introducido en Salida prueba: <b>PR-TEC-02</b></li> </ol>
<b>ENTRADA</b>	Salida prueba: <b>PR-TEC-02</b>
<b>SALIDA</b>	Command(s) completed successfully.
<b>ESTADO</b>	Aceptada

Tabla 125: Prueba PR-SOL-2

<b>ID</b>	PR-SOL-03
-----------	-----------

<b>OBJETIVO</b>	Validar que la salida obtenida en la prueba PR-TEC-03, es correcta según lenguaje SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Nombre base de datos igual que el introducido en Salida prueba: PR-TEC-03</li> </ol>
<b>ENTRADA</b>	Salida prueba: PR-TEC-03
<b>SALIDA</b>	Command(s) completed successfully.  <ol style="list-style-type: none"> <li>1. Insertados los primaryItems</li> </ol>
<b>ESTADO</b>	Aceptada

Tabla 126: Prueba PR-SOL-03

<b>ID</b>	PR-SOL-04
<b>OBJETIVO</b>	Validar que la salida obtenida en la prueba PR-TEC-04, es correcta según lenguaje SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Nombre base de datos igual que el introducido en Salida prueba: PR-TEC-04</li> </ol>
<b>ENTRADA</b>	Salida prueba: PR-TEC-04
<b>SALIDA</b>	Command(s) completed successfully.  <ol style="list-style-type: none"> <li>1. Insertados dimensions</li> <li>2. Insertados DimAttributes</li> <li>3. Insertados Relation_Dim_AttrDim</li> </ol>
<b>ESTADO</b>	Aceptada

Tabla 127: Prueba PR-SOL-04

<b>ID</b>	PR-SOL-05
<b>OBJETIVO</b>	Validar que la salida obtenida en la prueba PR-TEC-07, es correcta según lenguaje SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 1
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>3. Base de datos creada</li> <li>4. Nombre base de datos igual que el introducido en Salida prueba: PR-TEC-07</li> </ol>

<b>ENTRADA</b>	Salida prueba: PR-TEC-07
<b>SALIDA</b>	Incorrect syntax near 'use'.
<b>ESTADO</b>	Aceptada. No crea insert, no nombre base de datos.

Tabla 128: Prueba PR-SOL-05

### 6.2.2 Hito 2

<b>ID</b>	PR-SOL-06
<b>OBJETIVO</b>	Validar que la salida obtenida en la prueba PR-TEC-11, es correcta según lenguaje SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Nombre base de datos igual que el introducido en Salida prueba: <b>PR-TEC-11</b></li> <li>3. Inserciones de hito 1, en la base de datos creada</li> </ol>
<b>ENTRADA</b>	Salida prueba: <b>PR-TEC-11</b>
<b>SALIDA</b>	<p>Command(s) completed successfully.</p> <ol style="list-style-type: none"> <li>1. Insertada header</li> <li>2. Insertados Context</li> <li>3. Insertados Fact</li> <li>4. Insertado Unit</li> <li>5. Insertados Contexto_dimension_attrDim</li> </ol>
<b>ESTADO</b>	Aceptada

Tabla 129: Prueba PR-SOL-06

<b>ID</b>	PR-SOL-07
<b>OBJETIVO</b>	Validar que la salida obtenida en la prueba PR-TEC-12, es correcta según lenguaje SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Nombre base de datos igual que el introducido en Salida prueba: <b>PR-TEC-12</b></li> <li>3. Inserciones de hito 1, en la base de datos creada</li> </ol>
<b>ENTRADA</b>	Salida prueba: <b>PR-TEC-12</b>
<b>SALIDA</b>	<p>Command(s) completed successfully.</p> <ol style="list-style-type: none"> <li>1. No inserciones</li> </ol>
<b>ESTADO</b>	Aceptada

Tabla 130: Prueba PR-SOL-07

<b>ID</b>	PR-SOL-08
<b>OBJETIVO</b>	Validar que la salida obtenida en la prueba PR-TEC-13, es correcta según lenguaje SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Nombre base de datos igual que el introducido en Salida prueba: <b>PR-TEC-13</b></li> <li>3. Inserciones de hito 1, en la base de datos creada</li> </ol>
<b>ENTRADA</b>	Salida prueba: <b>PR-TEC-13</b>
<b>SALIDA</b>	<p>Command(s) completed successfully.</p> <ol style="list-style-type: none"> <li>6. Insertada header</li> <li>7. Insertados Context</li> </ol>
<b>ESTADO</b>	Aceptada

Tabla 131: Prueba PR-SOL-08

<b>ID</b>	PR-SOL-09
<b>OBJETIVO</b>	Validar que la salida obtenida en la prueba PR-TEC-17, es correcta según lenguaje SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 2
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>4. Base de datos creada</li> <li>5. Nombre base de datos igual que el introducido en Salida prueba: <b>PR-TEC-17</b></li> <li>6. Inserciones de hito 1, en la base de datos creada</li> </ol>
<b>ENTRADA</b>	Salida prueba: <b>PR-TEC-17</b>
<b>SALIDA</b>	Incorrect syntax near 'use'.
<b>ESTADO</b>	Aceptada

Tabla 132: Prueba PR-SOL-09

### 6.2.3 Hito 3

Las pruebas de la solución obtenida para este Hito 3, serán algo diferentes ya que en los anteriores hitos la solución son inserciones y en este hito la solución es un programa de validación SQL.

Hemos desarrollando dos fórmulas de validación, una para Solvency II y otra para FINREP 2012, para las pruebas tendremos unas pruebas generales de salida de las pruebas técnicas y otras

por cada fórmulas tratadas en diferentes apartados para una mayor organización en las que comprobemos la entrada de la base de datos y si la solución de validación es la correcta.

### 6.2.3.1 Pruebas generales Solución

<b>ID</b>	PR-SOL-10
<b>OBJETIVO</b>	Validar que la salida obtenida en la prueba PR-TEC-25, es correcta según lenguaje SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Nombre base de datos igual que el introducido en Salida prueba: <b>PR-TEC-25</b></li> <li>3. Inserciones de hito 1, en la base de datos creada</li> <li>4. Inserciones de hito 2, en la base de datos creada</li> </ol>
<b>ENTRADA</b>	Salida prueba: <b>PR-TEC-25</b>
<b>SALIDA</b>	Solución de validación por cada contexto o filtro.
<b>ESTADO</b>	Aceptada

Tabla 133: Prueba PR-SOL-10

<b>ID</b>	PR-SOL-11
<b>OBJETIVO</b>	Validar que la salida obtenida en la prueba PR-TEC-22, es correcta según lenguaje SQL.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Nombre base de datos igual que el introducido en Salida prueba: <b>PR-TEC-22</b></li> <li>3. Inserciones de hito 1, en la base de datos creada</li> <li>4. Inserciones de hito 2, en la base de datos creada</li> </ol>
<b>ENTRADA</b>	Salida prueba: <b>PR-TEC-22</b>
<b>SALIDA</b>	<p>Command(s) completed successfully.</p> <ol style="list-style-type: none"> <li>1. No validación</li> </ol>
<b>ESTADO</b>	Aceptada

Tabla 134: Prueba PR-SOL-11

<b>ID</b>	PR-SOL-12
<b>OBJETIVO</b>	Validar que la salida obtenida en la prueba PR-TEC-26, es correcta según lenguaje SQL.



<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	5. Base de datos creada 6. Nombre base de datos igual que el introducido en Salida prueba: <b>PR-TEC-26</b> 7. Inserciones de hito 1, en la base de datos creada 8. Inserciones de hito 2, en la base de datos creada
<b>ENTRADA</b>	Salida prueba: <b>PR-TEC-26</b>
<b>SALIDA</b>	Incorrect syntax near 'Use'.
<b>ESTADO</b>	Aceptada

Tabla 135: Prueba PR-SOL-12

### 6.2.3.2 Aserción AS17A de Solvency II

Para las pruebas de esta aserción primero veremos a que variables afecta, cual es la fórmula de validación y cuál es su precondition.

- Variables: AS17A, AS17 Y AS18
- Fórmula: El valor absoluto de AS17A- (AS17 + AS18) tiene que ser menor o igual a 3000
- Precondición: AS17 o AS1, al menos uno de los dos tiene que ser distinto de null.

Haremos nueve pruebas de esta validación con diferentes variables y diferentes resultados, para ello crearemos nueve contextos que serán:

1. Contexto con todas las variables y que cumplen la fórmula
2. Contexto con todas las variables y que no cumple la formula
3. Contexto con AS17 y AS18, que cumple la fórmula
4. Contexto con AS17, que cumple la fórmula
5. Contexto con AS17 y AS17A que no cumple la fórmula
6. Contexto con AS18 que no cumple la fórmula
7. Contexto con AS18 y AS17A que cumple la fórmula
8. Contexto con AS17A, que cumple la fórmula
9. Contexto con ninguna variable

<b>ID</b>	PR-SOL-13
<b>OBJETIVO</b>	Validar fórmula AS17A.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	1. Base de datos creada 2. Inserciones para prueba creadas
<b>ENTRADA</b>	Contexto con todas las variables y que cumplen la fórmula
<b>SALIDA</b>	Contexto1 VALIDACION CORRECTA

<b>ESTADO</b>	Aceptada
---------------	----------

Tabla 136: Prueba PR-SOL-13

<b>ID</b>	PR-SOL-14
<b>OBJETIVO</b>	Validar fórmula AS17A.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Inserciones para prueba creadas</li> </ol>
<b>ENTRADA</b>	Contexto con todas las variables y que no cumplen la fórmula
<b>SALIDA</b>	Contexto2 VALIDACION INCORRECTA
<b>ESTADO</b>	Aceptada

Tabla 137: Prueba PR-SOL-14

<b>ID</b>	PR-SOL-15
<b>OBJETIVO</b>	Validar fórmula AS17A.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Inserciones para prueba creadas</li> </ol>
<b>ENTRADA</b>	Contexto con AS17 y AS18, que cumple la fórmula
<b>SALIDA</b>	Contexto3 VALIDACION CORRECTA
<b>ESTADO</b>	Aceptada

Tabla 138: Prueba PR-SOL-15

<b>ID</b>	PR-SOL-16
<b>OBJETIVO</b>	Validar fórmula AS17A.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Inserciones para prueba creadas</li> </ol>
<b>ENTRADA</b>	Contexto con AS17, que cumple la fórmula
<b>SALIDA</b>	Contexto4 VALIDACION CORRECTA
<b>ESTADO</b>	Aceptada

Tabla 139: Prueba PR-SOL-16

<b>ID</b>	PR-SOL-17
<b>OBJETIVO</b>	Validar fórmula AS17A.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Inserciones para prueba creadas</li> </ol>
<b>ENTRADA</b>	Contexto con AS17 y AS17A, que no cumple la fórmula
<b>SALIDA</b>	Contexto5 VALIDACION INCORRECTA
<b>ESTADO</b>	Aceptada

Tabla 140: Prueba PR-SOL-17

<b>ID</b>	PR-SOL-18
<b>OBJETIVO</b>	Validar fórmula AS17A.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Inserciones para prueba creadas</li> </ol>
<b>ENTRADA</b>	Contexto con AS18, que no cumple la fórmula
<b>SALIDA</b>	Contexto6 VALIDACION INCORRECTA
<b>ESTADO</b>	Aceptada

Tabla 141: Prueba PR-SOL-18

<b>ID</b>	PR-SOL-19
<b>OBJETIVO</b>	Validar fórmula AS17A.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Inserciones para prueba creadas</li> </ol>
<b>ENTRADA</b>	Contexto con AS18 y AS17A, que cumple la fórmula
<b>SALIDA</b>	Contexto7 VALIDACION CORRECTA
<b>ESTADO</b>	Aceptada

Tabla 142: Prueba PR-SOL-19

<b>ID</b>	PR-SOL-20
<b>OBJETIVO</b>	Validar fórmula AS17A.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Inserciones para prueba creadas</li> </ol>
<b>ENTRADA</b>	Contexto con AS17A, que cumple la fórmula
<b>SALIDA</b>	Contexto8 PRECONDICION FALLIDA
<b>ESTADO</b>	Aceptada

Tabla 143: Prueba PR-SOL-20

<b>ID</b>	PR-SOL-21
<b>OBJETIVO</b>	Validar fórmula AS17A.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"> <li>1. Base de datos creada</li> <li>2. Inserciones para prueba creadas</li> </ol>
<b>ENTRADA</b>	Contexto con ninguna variable
<b>SALIDA</b>	Contexto9 PRECONDICION FALLIDA
<b>ESTADO</b>	Aceptada

Tabla 144: Prueba PR-SOL-21

#### 6.2.3.3 Aserción t03-hr01 de FINREP 2012

Para las pruebas de esta aserción veremos que filtros tiene y que valor tienen las variables de solución. Todas las variables tienen fallbackValue y no existen precondiciones

Por cada filtro las variables serán en hecho que resulte de la unión del filtro con:

- CarryingAmount → Dimensión AT, atributo de dimensión dAT:x3
- ImpairedNet → Dimensión AT, atributo de dimensión dAT:x9
- UnimpairedAndNotPastDue → Dim: AT y DL, atributos dAT:x20 y dTI:eq0d
- UnimpairedAndPastDueUnder90Days → Dim: AT y DL, atributos dAT:x20 y dTI:gt0d\_le90d
- UnimpairedAndPastDueBetween90and180Days → Dim: AT y DL, atributos dAT:x20 y dTI:gt90d\_le180d
- UnimpairedAndPastDueBetween180and1Year → Dim: AT y DL, atributos dAT:x20 y dTI:gt180d\_le1y
- UnimpairedAndPastDueOver1Year → Dim: AT y DL, atributos dAT:x20 y dTI:gt1y

Fórmula:

- $\text{CarryingAmount} = \text{UnimpairedAndNotPastDue} + \text{UnimpairedAndPastDueUnder90Days} + \text{UnimpairedAndPastDueBetween90and180Days} + \text{UnimpairedAndPastDueBetween180and1Year} + \text{UnimpairedAndPastDueOver1Year} + \text{ImpairedNet}$

Haremos dos pruebas para esta aserción dado que todas las variables tienen fallBackValue y no hay precondiciones como hemos dicho anteriormente:

1. Prueba que cumple la fórmula de validación
2. Pruebas que no cumple la fórmula de validación

<b>ID</b>	PR-SOL-22
<b>OBJETIVO</b>	Validar fórmula t03_hr01.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"><li>1. Base de datos creada</li><li>2. Inserciones para prueba creadas</li></ol>
<b>ENTRADA</b>	Valor de los hechos para que cumpla la validación
<b>SALIDA</b>	Validación Correcta
<b>ESTADO</b>	Aceptada

Tabla 145: Prueba PR-SOL-22

<b>ID</b>	PR-SOL-23
<b>OBJETIVO</b>	Validar fórmula t03_hr01.
<b>SUBSISTEMAS</b>	Subsistema de Hito 3
<b>PRECONDICIONES</b>	<ol style="list-style-type: none"><li>1. Base de datos creada</li><li>2. Inserciones para prueba creadas</li></ol>
<b>ENTRADA</b>	Valor de los hechos para que no se cumpla la validación
<b>SALIDA</b>	Validación Incorrecta
<b>ESTADO</b>	Aceptada

Tabla 146: Prueba PR-SOL-23

## 7. EVOLUCIÓN Y RESULTADOS

En este apartado trataremos la evolución del desarrollo técnico y los resultados obtenidos. Ya en el apartado de diseño con el apartado de pruebas se comenta que los resultados obtenidos son exitosos, en este apartado desarrollaremos este punto viendo como quedan las tablas tras las inserciones de las soluciones.

Igual que como en el resto del proyecto, se separará es apartando separando los hitos y dentro de esto en cada taxonomía, que son FINREP2012 y Solvency II.

### 7.1 Hito 1

Este hito ha sido implementado en el programa llamado “xbrl001.vbs”. Durante la evolución de este desarrollo se llegó a la versión número 3. En la primera versión se desarrollaron las inserciones para la taxonomía de FINREP 2012, en la segunda versión las inserciones para la taxonomía Solvency II, y más tarde se encontró un error en la realización de las inserciones de relación de dimensión con los atributos de dimensión y se creó su versión número 3.

Los resultados obtenidos para cada taxonomía con la versión 3 del programa (xbrl001\_v3.vbs) utilizando SQL Server Management Studio 2012, habiendo creado antes la base de datos y las tablas son los siguientes:

#### 7.1.1 FINREP 2012

De dicha inserción obtenemos las siguientes inserciones en cada una de las tablas:

Obtenemos 171 inserciones de atributos de dimensión (DimAttributes)

	ID	descr
1	dATx1	NULL
2	dATx10	NULL
3	dATx11	NULL
4	dATx12	NULL
5	dATx13	NULL
6	dATx14	NULL
7	dATx15	NULL
8	dATx16	NULL
9	dATx17	NULL
10	dATx18	NULL
11	dATx19	NULL
12	dATx2	NULL
13	dATx20	NULL
14	dATx3	NULL
15	dATx4	NULL
16	dATx5	NULL
17	dATx6	NULL
18	dATx7	NULL
19	dATx8	NULL
20	dATx9	NULL
21	dBAx1	NULL
22	dBAx2	NULL
23	dCDx1	NULL

Ilustración 29: FINREP2012 atributos de dimensión

Obtenemos 23 inserciones para la dimensión (Dimensions)

	ID	descr	Typ
1	AL	NULL	1
2	AS	NULL	1
3	AT	NULL	1
4	BT	NULL	1
5	CD	NULL	1
6	CI	NULL	1
7	CL	NULL	1
8	CR	NULL	1
9	CS	NULL	1
10	DL	NULL	1
11	EL	NULL	1
12	EQ	NULL	1
13	JI	NULL	1
14	LI	NULL	1
15	MA	NULL	1
16	OC	NULL	1
17	OM	NULL	1
18	PL	NULL	1
19	RI	NULL	1
20	RM	NULL	1
21	RP	NULL	1
22	RS	NULL	1
23	RT	NULL	1

Ilustración 30: FINREP2012 dimensiones

Obtenemos 17 inserciones para conceptos básicos (PrimaryItems)

	ID	descr	Tipo_Date
1	ad1	NULL	NULL
2	dd14	NULL	NULL
3	dd7	NULL	NULL
4	md10	NULL	NULL
5	md11	NULL	NULL
6	md12	NULL	NULL
7	md3	NULL	NULL
8	mi1	NULL	NULL
9	mi13	NULL	NULL
10	mi2	NULL	NULL
11	mi3	NULL	NULL
12	mi4	NULL	NULL
13	mi5	NULL	NULL
14	mi8	NULL	NULL
15	mi9	NULL	NULL
16	pi15	NULL	NULL
17	sd6	NULL	NULL

Ilustración 31: FINREP2012 conceptos básicos

Obtenemos 203 inserciones de relación entre dimensiones y atributos de dimensión(Relation\_Dim\_AttrDim)

	Dimension	Attrdim
1	AL	dCTx12
2	AL	dCTx23
3	AL	dCTx28
4	AL	dCTx38
5	AL	dCTx4
6	AS	dCTx1
7	AS	dCTx10
8	AS	dCTx11
9	AS	dCTx13
10	AS	dCTx14
11	AS	dCTx15
12	AS	dCTx16
13	AS	dCTx17
14	AS	dCTx18
15	AS	dCTx19
16	AS	dCTx2
17	AS	dCTx20
18	AS	dCTx22
19	AS	dCTx26
20	AS	dCTx27
21	AS	dCTx29
22	AS	dCTx3
23	AS	dCTx30

Ilustración 32: FINREP 2012 Relación Dim\_Atributo

7.1.2 Solvency II

De dicha inserción obtenemos las siguientes inserciones en cada una de las tablas de Solvency II:

Obtenemos 5 inserciones de atributos de dimensión (DimAttributes)

	ID	descr
1	per:AdHoc	NULL
2	per:Quarterly	NULL
3	per:Yearly	NULL
4	soc:Group	NULL
5	soc:Solo	NULL

Ilustración 33: Solvency II Atributos de dimensión

Obtenemos 2 inserciones para la dimensión (Dimensions)

	ID	descr	Type
1	PeriodicityDimension	NULL	1
2	SoloOrGroupDimension	NULL	1

Ilustración 34: Solvency II Dimensiones

Obtenemos 140 inserciones para conceptos básicos (PrimaryItems)



	ID	descr	Tipo_Date
1	a1	NULL	NULL
2	A10A	NULL	NULL
3	A10B	NULL	NULL
4	A11	NULL	NULL
5	A12	NULL	NULL
6	A13	NULL	NULL
7	A14	NULL	NULL
8	A14A	NULL	NULL
9	A16	NULL	NULL
10	A17	NULL	NULL
11	A18	NULL	NULL
12	A18A	NULL	NULL
13	A19	NULL	NULL
14	A19A	NULL	NULL
15	A2	NULL	NULL
16	A20	NULL	NULL
17	A21	NULL	NULL
18	A23	NULL	NULL
19	A25B	NULL	NULL
20	A26	NULL	NULL
21	A27	NULL	NULL
22	A29	NULL	NULL
23	A3	NULL	NULL
24	A30	NULL	NULL

Ilustración 35: Solvency II Conceptos básicos

Obtenemos 5 inserciones de relación entre dimensiones y atributos de dimensión(Relation\_Dim\_AttrDim)

	Dimension	Attrdim
1	PeriodicityDimension	per:AdHoc
2	PeriodicityDimension	per:Quarterly
3	PeriodicityDimension	per:Yearly
4	SoloOrGroupDimension	soc:Group
5	SoloOrGroupDimension	soc:Solo

Ilustración 36: Solvency II Relación Dim\_Atributo

## 7.2 Hito 2

Este hito ha sido implementado en el programa llamado “xbrl002.vbs”. Durante la evolución de este desarrollo se llegó a la segunda versión. En la primera versión se desarrollaron las inserciones para la taxonomía de FINREP 2012 y en la segunda versión las inserciones para la taxonomía Solvency II.

Los resultados obtenidos para cada taxonomía con la versión 2 del programa (xbrl002\_v2.vbs) utilizando SQL Server Management Studio 2012, habiendo creado antes la base de datos, las tablas y las inserciones del hito 1, son los siguientes:

7.2.1 FINREP 2012

De dicha inserción de la instancia obtenemos las siguientes inserciones en cada una de las tablas:

Obtenemos 4 inserciones de la cabecera (Header)

	lineNumber	line
1	1	<?xml version="1.0" encoding="UTF-8"?>
2	2	
3	3	<xbri:xbri xmlns:xsi="http://www.w3.org/2001/XMLSchema..."
4	4	<link:schemaRef xlink:type="simple" xlink:href="e...

Ilustración 37: FINREP Cabecera

Obtenemos 237 inserciones de contextos (Context)

	ID	entityID	instantPeriod	segment
1	e_x11_x3_emu_eur_x10_x1	abc	2011-06-12	NULL
2	e_x11_x3_emu_x2_x10_x1	abc	2011-06-12	NULL
3	e_x11_x3_eu_x10_x1	abc	2011-06-12	NULL
4	e_x11_x3_x10_x1	abc	2011-06-12	NULL
5	e_x11_x3_x2_eur_x10_x1	abc	2011-06-12	NULL
6	e_x11_x3_x2_x2_x10_x1	abc	2011-06-12	NULL
7	e_x11_x3_x4_x10_x1	abc	2011-06-12	NULL
8	e_x13_x3_emu_x14_eur_x16_x1	abc	2011-06-12	NULL
9	e_x13_x3_emu_x14_x2_x16_x1	abc	2011-06-12	NULL
10	e_x13_x3_eu_x14_x16_x1	abc	2011-06-12	NULL
11	e_x13_x3_x14_x16_x1	abc	2011-06-12	NULL
12	e_x13_x3_x2_x14_eur_x16_x1	abc	2011-06-12	NULL
13	e_x13_x3_x2_x14_x2_x16_x1	abc	2011-06-12	NULL
14	e_x13_x3_x4_x14_x16_x1	abc	2011-06-12	NULL
15	e_x17_x3_x21_x1	abc	2011-06-12	NULL
16	e_x17_x3_x3_x1	abc	2011-06-12	NULL
17	e_x17_x3_x6_x1	abc	2011-06-12	NULL
18	e_x19_x3_x21_x1	abc	2011-06-12	NULL
19	e_x19_x3_x3_x1	abc	2011-06-12	NULL
20	e_x19_x3_x6_x1	abc	2011-06-12	NULL
21	e_x22_x18_x10_x11_x1	abc	2011-06-12	NULL
22	e_x22_x18_x11_x1	abc	2011-06-12	NULL
23	e_x22_x18_x11_x11_x1	abc	2011-06-12	NULL
24	e_x22_x18_x12_x11_x1	abc	2011-06-12	NULL

Ilustración 38: FINREP contexto

Obtenemos 237 inserciones de hechos (Fact)

	id	baseID	decimals	contextRef	unitRef	value
1	1	mi1	0	e_x7_x20_x14_eq0d_x11_x1	EUR	5
2	2	mi1	0	e_x7_x20_x14_gt0d_le90d_x11_x1	EUR	5
3	3	mi1	0	e_x7_x20_x14_gt90d_le180d_x11_x1	EUR	5
4	4	mi1	0	e_x7_x20_x14_gt180d_le1y_x11_x1	EUR	5
5	5	mi1	0	e_x7_x20_x14_gt1y_x11_x1	EUR	5
6	6	mi1	0	e_x7_x20_x2_eq0d_x11_x1	EUR	1
7	7	mi1	0	e_x7_x20_x2_gt0d_le90d_x11_x1	EUR	1
8	8	mi1	0	e_x7_x20_x2_gt90d_le180d_x11_x1	EUR	1
9	9	mi1	0	e_x7_x20_x2_gt180d_le1y_x11_x1	EUR	1
10	10	mi1	0	e_x7_x20_x2_gt1y_x11_x1	EUR	1
11	11	mi1	0	e_x7_x20_x5_eq0d_x11_x1	EUR	1
12	12	mi1	0	e_x7_x20_x5_gt0d_le90d_x11_x1	EUR	1
13	13	mi1	0	e_x7_x20_x5_gt90d_le180d_x11_x1	EUR	1
14	14	mi1	0	e_x7_x20_x5_gt180d_le1y_x11_x1	EUR	1
15	15	mi1	0	e_x7_x20_x5_gt1y_x11_x1	EUR	1
16	16	mi1	0	e_x7_x20_x4_eq0d_x11_x1	EUR	1
17	17	mi1	0	e_x7_x20_x4_gt0d_le90d_x11_x1	EUR	1
18	18	mi1	0	e_x7_x20_x4_gt90d_le180d_x11_x1	EUR	1
19	19	mi1	0	e_x7_x20_x4_gt180d_le1y_x11_x1	EUR	1
20	20	mi1	0	e_x7_x20_x4_gt1y_x11_x1	EUR	1
21	21	mi1	0	e_x7_x20_x12_eq0d_x11_x1	EUR	1
22	22	mi1	0	e_x7_x20_x12_gt0d_le90d_x11_x1	EUR	1
23	23	mi1	0	e_x7_x20_x12_gt90d_le180d_x11_x1	EUR	1
24	24	mi1	0	e_x7_x20_x12_gt180d_le1y_x11_x1	EUR	1
25	25	mi1	0	e_x7_x20_x12_gt1y_x11_x1	EUR	1

Ilustración 39: FINREP Hechos

Obtenemos 1 inserciones para la unidad (Unit)

	id	measure
1	EUR	iso4217:EUR

Ilustración 40: FINREP unidad

Obtenemos 1278 inserciones para la relación de contexto con el atributo de dimensión y su dimensión (Context\_Dim\_AttrDim)

	contextID	dimension	attrDim
1	e_x11_x3_emu_eur_x10_x1	AS	dCTx11
2	e_x11_x3_emu_x2_x10_x1	AS	dCTx11
3	e_x11_x3_eu_x10_x1	AS	dCTx11
4	e_x11_x3_x10_x1	AS	dCTx11
5	e_x11_x3_x2_eur_x10_x1	AS	dCTx11
6	e_x11_x3_x2_x2_x10_x1	AS	dCTx11
7	e_x11_x3_x4_x10_x1	AS	dCTx11
8	e_x13_x3_emu_x14_eur_x16_x1	AS	dCTx13
9	e_x13_x3_emu_x14_x2_x16_x1	AS	dCTx13
10	e_x13_x3_eu_x14_x16_x1	AS	dCTx13
11	e_x13_x3_x14_x16_x1	AS	dCTx13
12	e_x13_x3_x2_x14_eur_x16_x1	AS	dCTx13
13	e_x13_x3_x2_x14_x2_x16_x1	AS	dCTx13
14	e_x13_x3_x4_x14_x16_x1	AS	dCTx13
15	e_x17_x3_x21_x1	AS	dCTx17
16	e_x17_x3_x3_x1	AS	dCTx17
17	e_x17_x3_x6_x1	AS	dCTx17
18	e_x19_x3_x21_x1	AS	dCTx19
19	e_x19_x3_x3_x1	AS	dCTx19
20	e_x19_x3_x6_x1	AS	dCTx19
21	e_x22_x18_x10_x11_x1	AS	dCTx22
22	e_x22_x18_x11_x1	AS	dCTx22
23	e_x22_x18_x11_x11_x1	AS	dCTx22
24	e_x22_x18_x12_x11_x1	AS	dCTx22
25	e_x22_x18_x2_x11_x1	AS	dCTx22

Ilustración 41: FINREP Relación contexto atributo de dimensión

## 7.2.2 Solvency II

De dicha inserción de la instancia obtenemos las siguientes inserciones en cada una de las tablas para Solvency II:

Obtenemos 4 inserciones de la cabecera (Header)

	lineNumber	line
1	1	<?xml version="1.0" encoding="UTF-8"?>
2	2	<!-- Generated by Fujitsu Interstage XWand B0177...
3	3	<xmlns:xbrl xmlns:xsi="http://www.w3.org/2001/XM...
4	4	<link:schemaRef xlink:type="simple" xlink:href="s...

Ilustración 42: Solvency II cabecera

Obtenemos 2 inserciones de contextos (Context)

	ID	entityID	instantPeriod	segment
1	Context_Instant_Quarterly_Solo	123456	2012-06-30	NULL
2	Context_Instant_Yearly_Solo	123456	2012-06-30	NULL

Ilustración 43: Solvency II contextos

Obtenemos 6 inserciones de Hechos (Fact)

	id	baseID	decimals	contextRef	unitRef	value
1	1	AS17A	0	Context_Instant_Quarterly_Solo	EURO	42000
2	2	AS18	0	Context_Instant_Quarterly_Solo	EURO	29655
3	3	AS17	0	Context_Instant_Quarterly_Solo	EURO	12345
4	4	AS18	0	Context_Instant_Yearly_Solo	EURO	69000
5	5	AS17	0	Context_Instant_Yearly_Solo	EURO	666
6	6	AS17A	0	Context_Instant_Yearly_Solo	EURO	100000

Ilustración 44: Solvency II hechos

Obtenemos 1 inserciones para la unidad (Unit)

	id	measure
1	EUR	iso4217:EUR

Ilustración 45: Solvency II unidad

Obtenemos 4 inserciones para la relación entre los contextos y los atributos de dimensión con sus dimensiones (Context\_Dim\_AttrDim)

	contextID	dimension	attrDim
1	Context_Instant_Quarterly_Solo	PeriodicityDimension	per:Quarterly
2	Context_Instant_Yearly_Solo	PeriodicityDimension	per:Yearly
3	Context_Instant_Quarterly_Solo	SoloOrGroupDimension	soc:Solo
4	Context_Instant_Yearly_Solo	SoloOrGroupDimension	soc:Solo

Ilustración 46: Solvency II Relación del contexto con los atributos de dimensión

## 7.3 Hito 3

### 7.3.1 Solvency II

De la validación de la taxonomía de acuerdo a la aserción AS17A, obtenemos el siguiente resultado, recordando que podíamos tener tres resultados diferentes, que la precondition no se cumple, que la formula de validación no sea correcta o que la validación sea correcta.

Messages
Context_Instant_Quarterly_Solo VALIDACION CORRECTA
Context_Instant_Yearly_Solo VALIDACION INCORRECTA

Ilustración 47: Validación Solvency II

En este resultado observamos que tenemos dos contextos, Context\_Instant\_Quarterly\_Solo y Context\_Instant\_Yearly\_Solo. Estos son los contextos de la taxonomía de Solvency II, que los podemos ver en los resultados del Hito2. Para cada contexto se ha ejecutado el fichero SQL de validación, y para el contexto Context\_Instant\_Quarterly\_Solo la validación fue correcta y para Context\_Instant\_Yearly\_Solo la validación no fue correcta. Esto quiere decir que los dos contextos cumplían la precondition pero que de los valores obtenidos de los Hechos solo el primer contexto a cumplido con la formula de validación de la aserción.

### 7.3.2 FINREP 2012

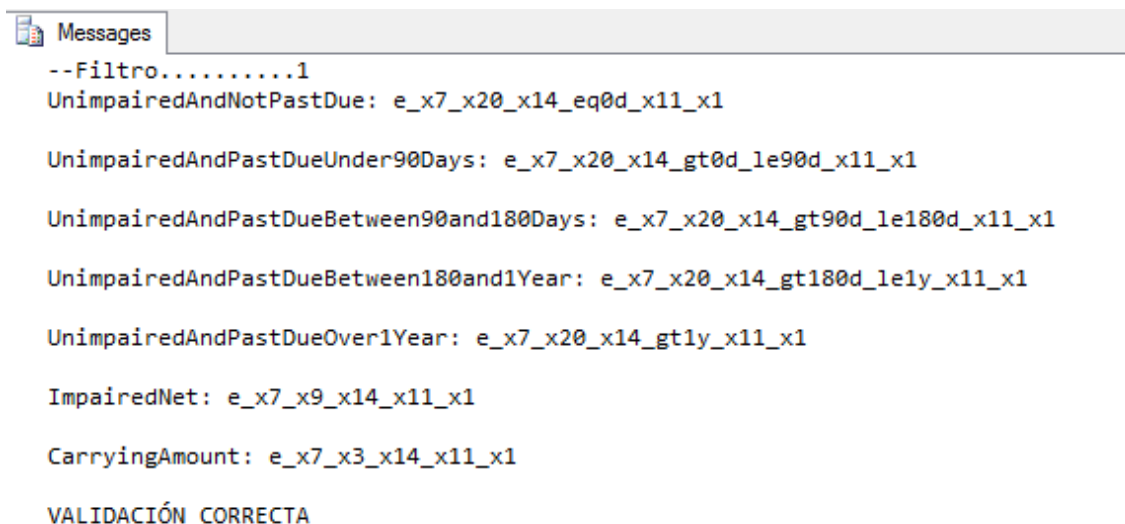
La validación de la taxonomía FINREP 2012 con respecto a la fórmula t003\_hr01, obtenemos los siguientes resultados, uno por cada filtro de la aserción. Se puede tener dos resultados diferentes, que la validación se cumple o que no se cumple.

El filtro 1 se forma con los siguientes elementos y su resultado es el siguiente:

Elementos:

- Concepto básico mi1
- Dimensión PL, atributo dPL:x11
- Dimensión RS, atributo dRS:x1
- Dimensión AS, atributo dCT:x7
- Dimensión CS, atributo dSE:x14

Resultado:



```
--Filtro.....1
UnimpairedAndNotPastDue: e_x7_x20_x14_eq0d_x11_x1

UnimpairedAndPastDueUnder90Days: e_x7_x20_x14_gt0d_le90d_x11_x1

UnimpairedAndPastDueBetween90and180Days: e_x7_x20_x14_gt90d_le180d_x11_x1

UnimpairedAndPastDueBetween180and1Year: e_x7_x20_x14_gt180d_le1y_x11_x1

UnimpairedAndPastDueOver1Year: e_x7_x20_x14_gt1y_x11_x1

ImpairedNet: e_x7_x9_x14_x11_x1

CarryingAmount: e_x7_x3_x14_x11_x1

VALIDACIÓN CORRECTA
```

Ilustración 48: Validación FINREP 2012 Filtro 1

El filtro 2 se forma con los siguientes elementos y su resultado es el siguiente:

Elementos:

- Concepto básico mi1
- Dimensión PL, atributo dPL:x11
- Dimensión RS, atributo dRS:x1

- Dimensión AS, atributo dCT:x22
- Dimensión CS, atributo dSE:x1

Resultado:

```
--Filtro.....2
UnimpairedAndNotPastDue: e_x7_x20_x14_eq0d_x11_x1

UnimpairedAndPastDueUnder90Days: e_x7_x20_x14_gt0d_le90d_x11_x1

UnimpairedAndPastDueBetween90and180Days: e_x7_x20_x14_gt90d_le180d_x11_x1

UnimpairedAndPastDueBetween180and1Year: e_x7_x20_x14_gt180d_le1y_x11_x1

UnimpairedAndPastDueOver1Year: e_x7_x20_x14_gt1y_x11_x1

ImpairedNet: e_x7_x9_x14_x11_x1

CarryingAmount: e_x7_x3_x14_x11_x1

VALIDACIÓN CORRECTA
```

Ilustración 49: Validación FINREP 2012 Filtro 2

El filtro 3 se forma con los siguientes elementos y su resultado es el siguiente:

Elementos:

- Concepto básico mi1
- Dimensión PL, atributo dPL:x11
- Dimensión RS, atributo dRS:x1
- Dimensión AS, atributo dCT:x3
- Dimensión CS, atributo dSE:x1

Resultado:

```
--Filtro.....3
UnimpairedAndNotPastDue: e_x7_x20_x14_eq0d_x11_x1

UnimpairedAndPastDueUnder90Days: e_x7_x20_x14_gt0d_le90d_x11_x1

UnimpairedAndPastDueBetween90and180Days: e_x7_x20_x14_gt90d_le180d_x11_x1

UnimpairedAndPastDueBetween180and1Year: e_x7_x20_x14_gt180d_le1y_x11_x1

UnimpairedAndPastDueOver1Year: e_x7_x20_x14_gt1y_x11_x1

ImpairedNet: e_x7_x9_x14_x11_x1

CarryingAmount: e_x7_x3_x14_x11_x1

VALIDACIÓN CORRECTA
```

Ilustración 50: Validación FINREP 2012 Filtro 3

## 8. PLANIFICACIÓN

En este apartado veremos la planificación de este proyecto, desde un punto de vista temporal. Partiendo de una estimación inicial y viendo como esta se va desarrollando a lo largo del proyecto.

Se considera útil utilizar una herramienta de gestión de proyectos, se ha utilizado en este caso Microsoft Project 2010, el cual en este caso nos proporciona un día de emiece de la tarea y un día de finalización de la tarea.

La estimación de horas de este proyecto, correspondientes a los 12 créditos ECTS que corresponde, son de 300 horas. Las horas han sido distribuidas de la siguiente manera:

- 60 horas para el estudio de nuevo conceptos relacionados con la especificación XBRL y de nuevas aplicaciones o lenguajes como son SQL Server, Arelle y VBS.
- Las otras restantes 240 horas han sido dedicadas íntegramente al análisis y desarrollo del proyecto, teniendo en cuenta que si la parte de estudio anterior no se hubiera podido realizar en este tiempo.

La duración en fecha de este proyecto será del 15 de mayo de 2012 hasta el 2 de septiembre de 2012.

- Las primeras 60 horas como veremos más tarde en la ilustración de las tareas del Project, han sido repartidas en 4 tareas, entre el 15 de mayo 2012 y el 29 de junio de 2012. Durante este periodo se estima una dedicación menor cada día por estar todavía en periodo lectivo, hasta hacer la suma de 60 horas, sabiendo que el mayor tiempo se dedicó a comprender el lenguaje y las taxonomías XBRL.
- Las segundas 240 horas han sido repartidas entre el 30 de junio 2012 y el 2 de septiembre de 2012. Durante este periodo se estima una dedicación de 4 horas diarias incluidos sábados y domingo, por estar en periodo vacacional.

Task Name	Start	Finish	Predecessors	% Complete
<input type="checkbox"/> Proyecto XBRL	Tue 15/05/12	Sun 02/09/12		100%
Inicio en XBRL	Tue 15/05/12	Fri 29/06/12		100%
Inicio en SQL Server	Tue 15/05/12	Fri 29/06/12		100%
Inicio en Arelle	Tue 15/05/12	Fri 29/06/12		100%
Inicio en Visual Basic Script	Tue 15/05/12	Fri 29/06/12		100%
<input type="checkbox"/> Análisis	Sat 30/06/12	Thu 12/07/12		100%
<input type="checkbox"/> Diseño	Sun 08/07/12	Wed 18/07/12		100%
<input type="checkbox"/> Implementación	Sun 15/07/12	Mon 27/08/12		100%
<input type="checkbox"/> Pruebas	Thu 26/07/12	Sun 02/09/12		100%
Valoración de resultados	Fri 31/08/12	Sun 02/09/12		100%

Ilustración 51: Tareas Gantt General



Task Name	Start	Finish	Predecessors	% Complete
<b>Análisis</b>	<b>Sat 30/06/12</b>	<b>Thu 12/07/12</b>		<b>100%</b>
Estudio y planificación del proyecto	Sat 30/06/12	Mon 02/07/12		100%
<b>Requisitos</b>	<b>Mon 02/07/12</b>	<b>Tue 10/07/12</b>		<b>100%</b>
Hito 1	Mon 02/07/12	Thu 05/07/12	7	100%
Hito 2	Thu 05/07/12	Sun 08/07/12	7	100%
Hito 3	Sun 08/07/12	Tue 10/07/12	7	100%
Casos de uso	Tue 10/07/12	Thu 12/07/12		100%
<b>Diseño</b>	<b>Sun 08/07/12</b>	<b>Wed 18/07/12</b>		<b>100%</b>
Selección de la Arquitectura	Sun 08/07/12	Wed 11/07/12		100%
Arquitectura de soporte	Wed 11/07/12	Wed 18/07/12		100%
Diseño físico de datos	Wed 11/07/12	Wed 18/07/12		100%
<b>Implementación</b>	<b>Sun 15/07/12</b>	<b>Mon 27/08/12</b>		<b>100%</b>
<b>Hito 1</b>	<b>Sun 15/07/12</b>	<b>Wed 25/07/12</b>	<b>9</b>	<b>100%</b>
Parte 1: FINREP 2012	Sun 15/07/12	Fri 20/07/12		100%
<b>Parte 2: Otras taxonomías</b>	<b>Fri 20/07/12</b>	<b>Wed 25/07/12</b>	<b>19</b>	<b>100%</b>
FINREP 2008	Fri 20/07/12	Sat 21/07/12		100%
Solvency II	Sun 22/07/12	Wed 25/07/12		100%
<b>Hito 2</b>	<b>Wed 25/07/12</b>	<b>Sun 12/08/12</b>	<b>10</b>	<b>100%</b>
FINREP 2012	Wed 25/07/12	Wed 01/08/12	19	100%
Solvency II	Tue 31/07/12	Sat 11/08/12	22	100%
<b>Hito 3</b>	<b>Mon 13/08/12</b>	<b>Mon 27/08/12</b>	<b>11</b>	<b>100%</b>
Solvency II	Mon 13/08/12	Wed 22/08/12	25	100%
FINREP 2012	Thu 23/08/12	Mon 27/08/12	24	100%

Ilustración 52: Tareas Gantt Parte 1

Task Name	Start	Finish	Predecessors	% Complete
<b>Pruebas</b>	<b>Thu 26/07/12</b>	<b>Sun 02/09/12</b>		<b>100%</b>
<b>Pruebas técnicas</b>	<b>Thu 26/07/12</b>	<b>Sun 02/09/12</b>		<b>100%</b>
Hito 1	Thu 26/07/12	Mon 30/07/12	18	100%
Hito 2	Mon 13/08/12	Fri 17/08/12	23	100%
Hito 3	Tue 28/08/12	Sun 02/09/12	26	100%
<b>Pruebas solución</b>	<b>Thu 26/07/12</b>	<b>Sun 02/09/12</b>		<b>100%</b>
Hito 1	Thu 26/07/12	Mon 30/07/12	18	100%
Hito 2	Mon 13/08/12	Fri 17/08/12	23	100%
Hito 3	Tue 28/08/12	Sun 02/09/12	26	100%
Valoración de resultados	Fri 31/08/12	Sun 02/09/12		100%

Ilustración 53: Tareas Gantt Parte 2

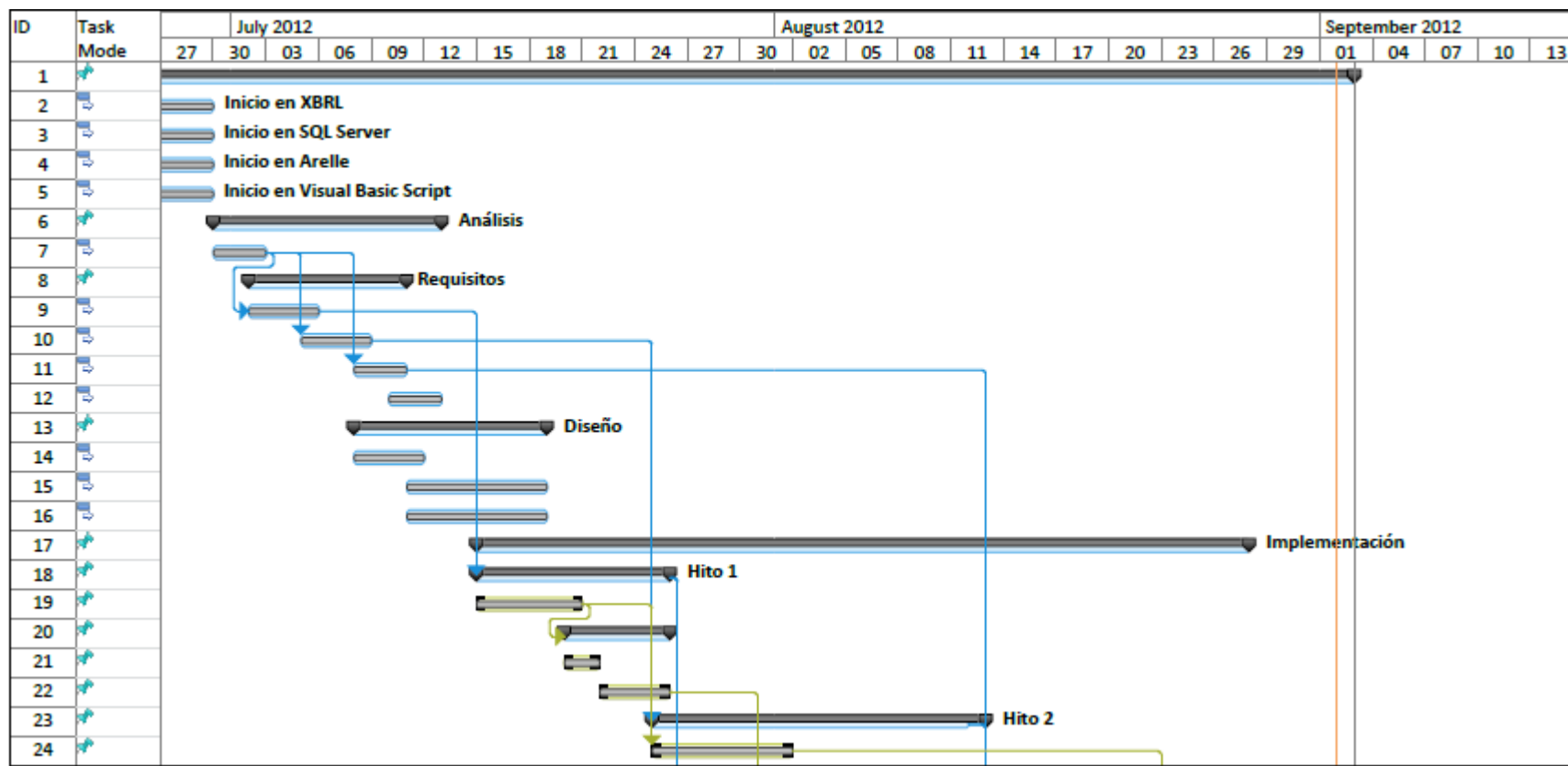


Ilustración 54: Diagrama de Gantt Parte 1

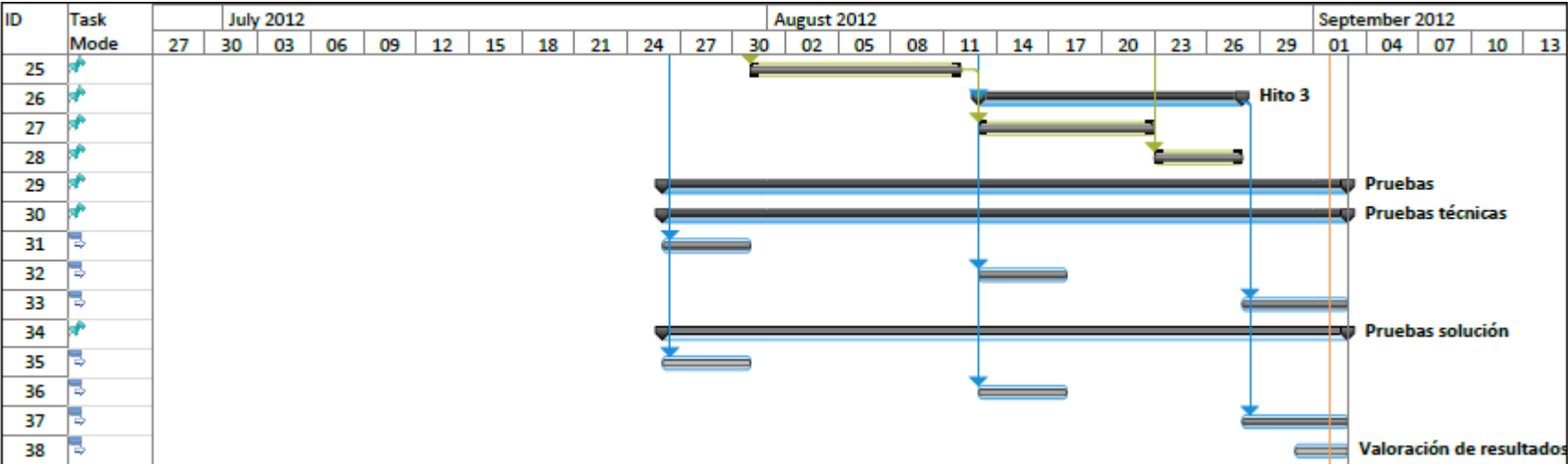


Ilustración 55: Diagrama de Gantt Parte 2

Project: Project1 Date: Sun 02/09/12	Task		Inactive Task		Start-only	
	Split		Inactive Milestone		Finish-only	
	Milestone		Inactive Summary		Deadline	
	Summary		Manual Task		Critical	
	Project Summary		Duration-only		Critical Split	
	External Tasks		Manual Summary Rollup		Progress	
	External Milestone		Manual Summary			

Ilustración 56: Diagrama de Gantt Leyenda

## 9. PRESUPUESTO

En este apartado veremos el presupuesto que se crea en el desarrollo de este proyecto, este presupuesto es importante para valorar el gasto que supondría si este proyecto estuviera dentro de una empresa.

En este presupuesto vamos a distinguir dos gastos, el gasto de personal, que será el gasto que se ocasiona por la dedicación del alumno, Yolanda León Román, y también existe el gasto de material, como puede ser ordenadores y software.

PRESUPUESTO DEL PROYECTO	COSTE POR UNIDAD	COSTE TOTAL RECURSO
<ul style="list-style-type: none"> <li><b>Personal</b> Yolanda León Román - 300 Horas</li> </ul>	<ul style="list-style-type: none"> <li><b>Personal</b> 30Euros/Hora</li> </ul>	<ul style="list-style-type: none"> <li><b>Personal</b> 9000 Euros</li> </ul>
<ul style="list-style-type: none"> <li><b>Material</b> Ordenador Windows 2007 64B SQL Server 2012  Project 2010  Visio 2010</li> </ul>	<ul style="list-style-type: none"> <li><b>Material</b> 1100 Euros  1500 Euros  <b>775 Euros</b>  <b>330 Euros</b></li> </ul>	<ul style="list-style-type: none"> <li>Material 183 Euros (Amort. 3 años)  250 Euros (Amort. 3 años)  130 Euros (Amort. 3 años)  55 Euros (Amort. 3 años)</li> </ul>
<b>TOTAL COSTE DEL PROYECTO</b>		<b>9618 Euros</b>

Tabla 147: Presupuesto del Proyecto

## 10. CONCLUSIONES Y TRABAJO FUTURO

En este apartado trataremos las conclusiones viendo y analizando los resultados y el camino que se ha seguido para llegar a ellos y el trabajo futuro a desarrollar una vez haya finalizado este proyecto.

Como conclusiones cabe destacar la automatización de dos taxonomías que son FINREP 2012 y Solvency II.

El primer paso de este proyecto para cada taxonomía era la transformación de los datos y metadatos de la taxonomía a una base de datos, en este caso una base de datos en SQL Server, y el segundo paso es la validación de la taxonomía mediante las aserciones.

El primer paso que es desarrollando en los Hitos 1 y 2, ha sido completamente satisfactorio, ya que todos los datos contenidos en las taxonomías ahora los tenemos en una base de datos relacional en SQL, a la que podemos acceder de manera rápida.

El segundo paso que sería el desarrollo del hito 3 que consiste en que una vez extraídos los datos de la taxonomía a SQL Server validar estos datos de acuerdo a las aserciones de la taxonomía insertada. Esta validación es una parte importante de los informes XBRL ya que el interés de estos informes en XBRL consiste en la validación de los datos en origen y en destino. Para la taxonomía de Solvency II se ha conseguido un resultado satisfactorio en su fichero de aserciones y su taxonomía ha sido validada, para la taxonomía FINREP 2012, como se ha comentado en el apartado de implementación, ha surgido un problema con el tratamiento del fichero de aserciones desde Arelle, y por eso sólo se ha podido desarrollar la primera validación. El desarrollo de esta validación ha sido exitoso y la taxonomía ha quedado validada para esta fórmula de aserción.

En cuanto a las conclusiones de lo aprendido en el proyecto y de los retos que han surgido puedo decir que he aprendido varios lenguajes de programación como son VBS y SQL Server y al ser lenguajes nuevos para mí a veces han surgido contratiempos arreglados con los tutoriales. Pero sin duda el reto más grande ha sido la misma especificación XBRL, el lenguaje basado en XML, XBRL es un lenguaje por ser de XML sencillo, pero el problema ha surgido en la terminología económica y financiera de los mismos creadores de la especificación XBRL, es un lenguaje pensado para economistas y por economistas, y esto en ciertos momentos puede ser una dificultad añadida. Otro de los retos y cosas que he aprendido durante el proyecto ha sido las bases de datos multidimensionales, no planificadas en el plan de estudios del Grado en Ingeniería Informática.

El trabajo futuro concluido este proyecto es en sí la publicación de la solución de este proyecto en la página <http://www.openfiling.info> una comunidad de software abierto para la utilización de este estándar, XBRL, donde ya se han ido publicando partes de la tesis doctoral de la que forma parte este proyecto. Como parte de esta publicación y antes de poder publicar, el trabajo futuro será el desarrollo de todas las aserciones para la taxonomía de FINREP 2012, investigando las diferentes alternativas con el problema encontrado.

Otra parte del trabajo futuro es la automatización de más taxonomías XBRL.

## 11. RESUMEN

Este trabajo consiste en la transformación a un lenguaje gestor de base de datos unos documentos instancias y en sí unas taxonomías XBRL. XBRL (*extensible Business Reporting Language*), es un lenguaje basado en XML que sirve para hacer automático el intercambio de información financiera.

El desarrollo de este proyecto surge gracias a varias directivas europeas que entraran en vigor en el año 2014 para el envío de informes económicos XBRL por parte de bancos y empresas a las entidades financieras. Sumado a esto existe el problema de que existen muy pocos profesionales que sepan tratar informes XBRL y la no existencia de un programa que mapee estos informes XBRL a un programa fácil de usar.

Por tanto, el objetivo primordial de este proyecto es la transformación de taxonomías XBRL a Base de datos SQL, se elige SQL puesto que se considera un lenguaje conocido para la mayoría de los bancos centrales, para esta transformación se diferencias tres hitos y cada uno de ellos tiene el siguiente objetivo:

1. Primer Hito: Creación de inserciones de fichero Arelle. Mediante el lenguaje Visual Basic Script obtener los metadatos y así la creación de los conceptos básicos, dimensiones, atributos y relación dimensión-atributo.
2. Segunda Fase: Creación de inserciones de fichero instancia XBRL. Mediante el lenguaje Visual Basic Script acceder al fichero instancia XBRL, y extraer los metadatos como son los contextos, los hechos, y la relación de los contextos con las dimensiones.
3. Tercera Fase: Tratamiento de fórmulas y validaciones. Mediante el lenguaje Visual Basic Script generar un programa de Transact SQL que desarrolle la aserción de la taxonomía para analizar y validar los datos introducidos en la base de datos.

El análisis y desarrollo de este proyecto se lleva a cabo separando Hitos, ya que para implementar el desarrollo de un hito no es necesario ninguno de los otros dos. El resultado al ejecutar la implementación de los hitos será lo que tiene correspondencia y necesidad de los otros ya que la solución del primer hito genera unas sentencias de inserciones SQL sin las cuales las inserciones de la solución del Hito 2 no se podría ejecutar y la solución del hito 3 es la validación de las taxonomías que sin las inserciones de los hitos 1 y 2 no tiene sentido realizar, puesto que la base de datos estará vacía.

En este proyecto se pretende la automatización de las taxonomías de manera que cualquier taxonomía puede ser ejecutada con la solución de este proyecto y así obtener los datos en SQL. Durante el estudio a la hora de implementar la solución se encuentra el problema con la taxonomía de FINREP 2008 ya que esta sigue un formato anterior a las otras dos taxonomías probadas que son FINREP 2012 y Solvency II y puesto que la taxonomía de FINREP 2008 es más antigua, se decide crear los programas de este proyecto para FINREP 2012 y Solvency II.

Durante el desarrollo de los hitos 1 y 2 no encontramos ningún problema y todos los datos de la taxonomía son introducidos en los ficheros para la inserción en la base de datos.

Durante el desarrollo del hito 3 se encuentra un problema en la taxonomía de FINREP 2012 generado por el programa que se utiliza durante este proyecto llamado Arelle para pasar los ficheros XML y XBRL a texto plano para tratarlos con mayor facilidad. Este problema consiste en la generación incorrecta del archivo de aserciones faltando metadatos de los atributos de dimensión. Encontrándonos con este problema se decide tratar una de las aserciones de la taxonomía FINREP 2012, para crear su código SQL y así validar la taxonomía con al menos esta aserción.

Los resultados de este proyecto son los esperados si quitamos ese problema ajeno a nosotros en las aserciones de FINREP 2012.

La planificación ha sido cumplida según la estimación inicial.

En cuanto al trabajo futuro acabado este proyecto, consiste en la generación de las aserciones de FINREP 2012 correctamente para poder subir el código generado a <http://www.openfiling.info/> para que llegue al mayor número de gente interesada. Y otra parte del trabajo futuro será la automatización de más taxonomías XBRL.

Resumen en inglés:

This work consists of mapping XBRL documents instances in databases and converse. XBRL (eXtensible Business Reporting Language) is an XML-based language used to make automatic exchange of financial information.

The development of this project came about thanks to various European directives came into force in 2014 for sending XBRL financial reports by banks and companies to financial institutions. Added to this there is the problem that there are very few professionals who can treat XBRL reports and the absence of a program that maps these XBRL reports that they are not easy programs to use. Therefore, the primary objective of this project is the transformation of XBRL taxonomies to SQL Database, SQL is chosen because it is considered a language known to most central banks and regulators, for this transformation are three milestones differences and each they have the following objective.

This project aims to automate taxonomies so that any taxonomy can be run with the solution of this project and get the data into SQL. During the study at the time of implementing the solution is the problem with the taxonomy of FINREP 2008 since this is an older format to the other two tested taxonomies are FINREP 2012 and Solvency II and since 2008. FINREP taxonomy is old, decided to create this project programs for FINREP 2012 and Solvency II.

The results of this project are expected. Planning has been fulfilled as the initial estimate.

As future work finishing this project, is generating assertions correctly FINREP 2012 to upload the code generated <http://www.openfiling.info/> to reach the greatest number of people interested. And part of the future work will be more automation XBRL taxonomies.





## 12. ANEXOS

### 12.1 GLOSARIO DE TÉRMINOS

Ordenadas por nombre

<b>Atributo de Dimensión</b>	Metadato expresado en una tabla en el modelo de base de datos
<b>Base de datos</b>	Base de datos en la que queremos crear nuestra taxonomía en SQL.
<b>BCE</b>	Banco central europeo
<b>BIS</b>	Banco Internacional de pago
<b>Código abierto</b>	Software distribuido y desarrollado libremente
<b>Conceptos Básicos</b>	Metadato expresado en una tabla en el modelo de base de datos como primary Items
<b>Contexto</b>	Metadato expresado en una tabla en el modelo de base de datos.
<b>Dimensión</b>	Metadato expresado en una tabla en el modelo de base de datos
<b>Eurostat</b>	Oficina de estadísticas de la Comisión Europea que produce los datos de la UE y promueve los métodos estadísticos.
<b>Fichero aserción</b>	Fichero de aserciones de la taxonomía pasado por Arelle. Para la taxonomía FINREP 2012 este fichero corresponde solo a la fórmula t02_h01 modificado para añadir los atributos de dimensión.
<b>Fichero Dimensión</b>	Fichero obtenido por <b>Arelle</b> de la instancia de la taxonomía que contiene los <b>atributos, dimensiones y atributos de dimensión</b> .
<b>Fichero Inserciones</b>	Fichero en el que se guardan las inserciones de SQL para los datos del <b>Fichero Dimensión</b> .
<b>Fichero Log</b>	Fichero en el que se guarda información relativa a la ejecución del programa
<b>Fichero Procedure</b>	Fichero SQL de validación. Solución del H3.
<b>Fichero Validation</b>	Fichero opcional de uso para información de validaciones.
<b>Filtros</b>	La unión de conceptos básicos con dimensiones y atributos de dimensión que marcan la manera que sigue una validación.
<b>Hecho</b>	Metadato expresado en una tabla en el modelo de base de datos como Facts
<b>Instancia XBRL</b>	Documento XML donde en el que se encuentran etiquetados cada valor del negocio asociado a un contexto y una unidad de medida. Una misma instancia puede tener varias entidades y varios periodos.
<b>IRFS</b>	Normas Internacionales de Información Financiera
<b>Metadato</b>	Datos o etiquetas que definen información de

	los propios datos
<b>NIIF</b>	IRFS
<b>OCDE</b>	Organismo para la Cooperación y el Desarrollo Económico
<b>SDMX</b>	Intercambio de Datos y Metadatos, por la traducción de sus siglas a castellano.
<b>SGBD</b>	Sistema Gestor de Base de Datos
<b>SGML</b>	Estándar de lenguaje Marcado Generalizado
<b>SQL</b>	Es un lenguaje declarativo de acceso a bases de datos relacionales en la que se ejecutará en este proyecto.
<b>Taxonomía</b>	Son los diccionarios del lenguaje XBRL, se trata de esquemas de clasificación que definen etiquetas específicas para cada elemento de específico de información
<b>UE</b>	Unión Europea
<b>Usuario</b>	Persona que interactúa con el sistema.
<b>VBS</b>	Visual Basic Script, lenguaje de Microsoft en que se desarrollaran los programas en este proyecto.

Tabla 148: Glosario de Términos

## 12.2 BIBLIOGRAFIA

- 1.- Arelle ([www.arelle.org](http://www.arelle.org))
- 2.- World Wide Web Consortium ([www.w3.org](http://www.w3.org))
- 3.- Hoffman, Charles. What is XBRL? Ed. Mayo 2008
- 4.- Asociación XBRL en España ([www.xbrl.es](http://www.xbrl.es))
- 5.- Banco de España. Diseño técnico de FINREP 2008  
([http://www.bde.es/es/fr/documentacion/taxonomias/es-be-FINREP/6-2008/2008-11-26/es-be-FINREP\\_tecnico.pdf](http://www.bde.es/es/fr/documentacion/taxonomias/es-be-FINREP/6-2008/2008-11-26/es-be-FINREP_tecnico.pdf))
- 6.- EIOPA (<https://eiopa.europa.eu/>)
- 7.- EUROFILING (<http://www.eurofiling.info/FINREPTaxonomy/taxonomy2012.shtml>)
- 9.- BIS ([www.bis.org](http://www.bis.org))
- 10.- BCE ([www.ecb.int](http://www.ecb.int))
- 11.- OCDE ([www.ocde.org](http://www.ocde.org))
- 12.- Eurostat (<http://epp.eurostat.ec.europa.eu>)
- 13.- SDMX ([www.sdmx.org](http://www.sdmx.org))
- 14.- IRFS (<http://www.ifrs.org>)
- 15.- Libro Blanco XBRL. XBRL España ([http://www.xbrl.es/downloads/libros/Libro\\_Blanco.pdf](http://www.xbrl.es/downloads/libros/Libro_Blanco.pdf))

## 12.3 TAXONOMIAS

### 12.3.1 FINREP 2008

FINREP 2008 es una taxonomía de XBRL, diseñada por el Banco de España. Esta taxonomía debe ser utilizada por las Entidades de Crédito para el envío al Banco de España, de la declaración de sus Estados.

El objetivo de esta taxonomía es el modelado de los estados públicos y reservados del informe individual y consolidado, con lo cual los estados se dividen en individuales y consolidados y cada uno de ellos formará un módulo de taxonomía diferente. Dentro de estos módulos la taxonomía está dividida en informes.

Esta taxonomía está basada en las normas NIC que están implícitas en las Normas Internacionales de Información Financiera (NIIF) o también llamadas IFRS<sub>14</sub> por sus siglas en inglés.

### 12.3.2 FINREP 2012

Esta taxonomía empezó en abril de 2012 cuando el Banco de España se embarcó en un proyecto para desarrollar una taxonomía de XBRL, que refleja la información estadística recogida por las organizaciones de supervisión bancaria para el Banco Central Europeo.

Esta taxonomía al igual que la anterior, como es FINREP pero en otra versión también está basada en las mismas normas.

### 12.3.3 Solvency II

Solvency II es una taxonomía de XBRL, creada por la Autoridad Europea de Seguros y Pensiones de Jubilación (EIOPA).

El proyecto Solvency II tiene como objetivo revisar el régimen prudencial de seguros y de reaseguros en la Unión Europea. Como primer paso, la Directiva Solvency II fue aprobada por el Consejo de la Unión Europea y el Parlamento en noviembre de 2009.

Esta taxonomía se aplicará en 2014 a entre cuatro mil y cinco mil de las más grandes compañías de seguros de diferentes países de la Unión Europea.